Automated Software Engineering manuscript No.
(will be inserted by the editor)

How to Certify Machine Learning Based Safety-critical Systems?

A Systematic Literature Review

Florian Tambon[1] · Gabriel Laberge[1] · Le An[1] · Amin Nikanjam[1] · Paulina Stevia Nouwou Mindom[1] · Yann Pequignot[2] · Foutse Khomh[1] · Giulio Antoniol[1] · Ettore Merlo[1] · François Laviolette[2]

Received: date / Accepted: date

Abstract *Context:* Machine Learning (ML) has been at the heart of many innovations over the past years. However, including it in so-called "safety-critical" systems such as automotive or aeronautic has proven to be very challenging, since the shift in paradigm that ML brings completely changes traditional certification approaches.

Objective: This paper aims to elucidate challenges related to the certification of ML-based safety-critical systems, as well as the solutions that are proposed in the literature to tackle them, answering the question "How to Certify Machine Learning Based Safety-critical Systems?".

Method: We conduct a Systematic Literature Review (SLR) of research papers published between 2015 to 2020, covering topics related to the certification of ML systems. In total, we identified 229 papers covering topics considered to be the main pillars of ML certification: *Robustness*, *Uncertainty*, *Explainability*, *Verification*, *Safe Reinforcement Learning*, and *Direct Certification*. We analyzed the main trends and problems of each sub-field and provided summaries of the papers extracted.

Results: The SLR results highlighted the enthusiasm of the community for this subject, as well as the lack of diversity in term of datasets and type of ML models. It also emphasized the need to further develop connections between academia and industries to deepen the domain study. Finally, it also illustrated

This work is supported by the DEEL project CRDPJ 537462-18 funded by the National Science and Engineering Research Council of Canada (NSERC) and the Consortium for Research and Innovation in Aerospace in Québec (CRIAQ), together with its industrial partners Thales Canada inc, Bell Textron Canada Limited, CAE inc and Bombardier inc.

[1] Polytechnique Montréal, Québec, Canada
E-mail: {florian-2.tambon, gabriel.laberge, le.an, amin.nikanjam, paulina-stevia.nouwou-mindom, foutse.khomh, giuliano.antoniol, ettore.merlo}@polymtl.ca
[2] Laval University, Québec, Canada
E-mail: francois.laviolette@ift.ulaval.ca, yann.pequignot@iid.ulaval.ca

the necessity to build connections between the above mention main pillars that are for now mainly studied separately.

Conclusion We highlighted current efforts deployed to enable the certification of ML based software systems, and discuss some future research directions.

Keywords Machine Learning, Certification, Safety-critical, Systematic Literature Review

1 Introduction

Machine Learning (ML) is drastically changing the way we interact with the world. We are now using software applications powered by ML in critical aspects of our daily lives; from finance, energy, to health and transportation. Thanks to frequent innovations in domains like Deep Learning (DL) and Reinforcement Learning (RL), the adoption of ML is expected to keep rising and the economic benefits of systems powered by ML is forecast to reach 30.6 Billions $ by 2024[1]. However, the integration of ML in systems is not without risks, especially in *safety-critical* systems such as avionic or automotive, where any mistake can lead to catastrophic events[2]. Therefore, before applying and deploying any machine learning based components into a safety-critical system, these components need to be certified.

Certification of systems aims at ensuring their reliability and safety through the definition of a clear standard encompassing method such as safety-assurance, verification or validation that can be assessed by an independent authority. This certification aspect is even more important in the case of safety-critical systems. If this certification aspect is already preponderant for mechanical systems, it has also been a mandatory step for the inclusion of software or electronic related component, especially with the rise of embedded software. As such, standards were developed in order to tackle this challenge: IEC 61508[3] developed as an international standard to certify electrical, electronic and programmable electronic safety related systems with ISO 26262[4] being the adaptation (and improvement) of this standard applied specifically to road vehicles safety or DO-178 C[5][260] created for the certification of airborne systems and equipment, introducing for instance MC/DC coverage criteria as a requirement. Those standards generally aim to deal with the functional safety consideration of a safety-critical system providing guidelines, risk-level and a range of requirements that need to be enforced in order to reach said level based on the criticality of the system and/or component inside the whole architecture. Those standards are now widely used in those safety-critical systems.

1 `https://www.forbes.com/sites/louiscolumbus/2020/01/19/roundup-of-machine-learning-forecasts-and-market-estimates-2020/`

2 `https://www.cbc.ca/news/business/uber-self-driving-car-2018-fatal-crash-software-flaws-1.5349581`

3 `https://webstore.iec.ch/publication/6007`

4 `https://www.iso.org/standard/68383.html`

5 `https://my.rtca.org/NC__Product?id=a1B36000001IcmqEAC`

However, the introduction of machine learning components in those systems changes the game completely: while ML can be extremely useful as it conveys the promise of replicating human knowledge with the power of a machine, it also induces a shift in software development and certification practices. Traditionally, software systems are constructed deductively, by writing down the rules that govern the behavior of the system as program code. With ML techniques, these rules are generated in an inductive way (i.e., *learned*) from training data. With this paradigm shift, the notion of specifications is transposed from the code itself to the data and the learning process, as a consequence, most of previously defined standards are not applicable. In the light of this observation, the scientific community has been working to define new standards specific to the unique nature of machine learning applications (see **Section 2**).

This paper aims to provide a snapshot of the current answers to the question "How to certify machine learning based safety-critical systems?" by examining the progress made over the past years with an emphasis on the transportation domain. In particular, we want to shed light on recent advances in different fields of machine learning regarding this question. We also want to identify current gaps and promising research avenues that could help to reach the community's goal of certifying ML based safety-critical systems. To achieve this objective, we adopt the Systematic Literature Review (SLR) approach, which differs from traditional literature reviews, by its rigour and strict methodology that aim to eliminate biases when gathering and synthesizing information related to a specific research question [122].

The paper remainder of this paper is structured as follows. **Section 2** discusses key concepts related to the content of the paper, **Section 3** describes in depth the SLR methodology; explaining how we have applied it to our study. **Section 4** presents descriptive statistics about our collected data and a taxonomy derived from the studied papers. In **Section 5**, we leverage the collected data and present a broad spectrum of the progresses made and explain how they can help certification. We also discuss current limitations and possible leads. We subdivide this section as follow; **Section 5.1** is about *Robustness*, **Section 5.2** is about *Uncertainty* and *Out-of-Distribution* (OOD), **Section 5.3** is about *Explainability* of models, **Section 5.4** is about *Verification* methods which are divided between *Formal* and *Non-Formal* (Testing) ones, **Section 5.5** is about *Safety considerations in Reinforcement Learning* (RL), **Section 5.6** deals with *Direction Certifications* proposals, and **Section 5.7** references all other papers not falling in the previous categories. **Section 6** summarizes key lessons derived from the review. Finally **Section 7**, concludes the paper.

2 Background

ML is a sub-domain of Artificial Intelligence (AI) that makes decisions based on information learned from data [263]. The "learning" part is very important, as it introduces a major difference from classical software where the logic is coded by a human; this paradigm shift is the quintessential issue of using ML in safety-critical systems as it cannot comply with the drastic security measures needed. ML is generally divided into three main categories; *supervised* learning, *unsupervised* learning, and *reinforcement* learning. The difference between those denominations comes from the way the model actually processes the data. In supervised learning, the model is given data along with a label (for example, a picture of a dog with the label "dog") and has to determine a boundary in between the data; such as all data with the same label are inside the same group (*i.e.*, class). In unsupervised learning, the model doesn't have access to labels and has to learn to make the same prediction for similar instances (*e.g.* clusters) in the data. In reinforcement learning, the model evolves in an environment and reacts to it based on reward feedback, so there is no fixed dataset as the data is being generated continuously from interaction with the environment.

Safety certification of software and/or electronic component is not new: the first version of IEC 61508 was released around 1998 to tackle those considerations and was later upgraded to match the evolution of the state-of-the-art. However, it is only recently that the specificity of machine learning was acknowledged with the improvement of the techniques and the increased usage of the paradigm as a component in bigger systems, with for instance EASA (European Union Aviation Safety Agency) releasing a Concept Paper "First usable guidance for Level 1 machine learning applications"[6] following its AI Roadmap. Nonetheless, to this day, there are no released standards that tackle specifically machine learning certification.

ISO 26262 Road vehicles — Functional safety [109], tackles issue of safety-critical systems that contain one or more electronic/electrical components in passenger cars. They analyze hazard due to malfunction of such components or their interaction and discuss how to mitigate them. It was designed to take into account the addition of ADAS (Advanced Driver Assistance System), yet it doesn't acknowledge the specificity of machine learning. In the same train of thoughts but for aeronautics, DO-178C is the primary document used for certification of airborne systems and equipment. Those two standards are based on risk-level and a range of criteria that needs to be matched to reach a certain risk-level that has to be correlated to the impact of the component on the system and the consequences in case of malfunction.
ISO/PAS 21448 Safety of the intended functionality (SOTIF)[211] is a recent standard that took a different approach: instead of tackling functional safety, this one aims for *intended* functionality, that is when in absence of fault in the program (in the traditional sense), an unexpected behavior is observed. Indeed, a hazard can happen even without a system failure and this standard attempts to take into consideration the impact of an unknown scenario that the system was not exactly built to tackle. It was also designed to be applied for

6 https://www.easa.europa.eu/newsroom-and-events/news/easa-releases-consultation-its-first-usable-guidance-level-1-machine

instance to Advanced Driver Assistance System (ADAS) and brings forward an important reflection about machine learning specification: our algorithm can work as it was trained to but not as we intended to. Yet, the standard does not offer strict criteria that could be correlated to problems affecting machine learning algorithms such as robustness against out-of-distribution, adversarial examples, or uncertainty.

Recently, it seems the ISO organization is working towards more machine learning/artificial intelligence oriented standards: in particular, ISO Technical Committee dealing with Artifical Intelligence (ISO/IEC JTC 1/SC 42)[7] lists to this date 7 published standards dedicated to Artificial Intelligence subjects and 22 more are under development. One example is ISO/IEC 20546:2019[8] which defines a common vocabulary and provides an overview of the Big Data Field. Another example is ISO/IEC TR 24028:2020 [9] which gives an overview of trustworthiness in AI or ISO/IEC TR 24029-1:2021[10] which provides background about robustness in AI. All the above mentioned papers only offer an overview of possible directions without proposing detailed specification/risk levels for each critical property of ML systems. The situation could be improved with the advent of standards such as ISO/IEC AWI TR 5469 Artificial intelligence — Functional safety and AI systems[11] which seem to be in direct relation with ISO 26262; however not much information is available at the moment, as they are still under development.

In parallel to these standards, the scientific community and various organizations have tried to come up with ways to characterize the paradigm shift that ML induces to systems and to tackle these unique issues. In [86], the author aim to certify the level of quality of a data repository with examples of data quality from three organizations using SQL databases : They use as a base, the ISO/IEC 25012 which defines data quality and characteristics and ISO/IEC25024 which bridges those concepts of data quality with "quality property" in order to evaluate data quality. The FAA introduced the notion of Overarching Properties[12] as an alternative to DO-178C that could fit better for ML components: based on 3 main sufficient properties: intent ("the defined intended behavior is correct and complete with respect to the desired behavior"), correctness ("the implementation is correct with respect to its defined intended behavior under foreseeable operating conditions"), and acceptability ("any part of the implementation that is not required by the defined intended behavior has no unacceptable safety impact"), which can be summed up as follows: be specified correctly (intent), do the right things (correctness) and do no wrong (acceptability). However, as there is no clear checklist here on

[7] https://www.iso.org/committee/6794475/x/catalogue/
[8] https://www.iso.org/standard/68305.html?browse=tc
[9] https://www.iso.org/standard/77608.html?browse=tc
[10] https://www.iso.org/standard/77609.html?browse=tc
[11] https://www.iso.org/standard/81283.html?browse=tc
[12] https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/media/TC_Overarching.pdf

what criteria to comply to, it is more complicated to enforce in practice. As of now, the approach does not seem to have been adopted widely.

To help tackle scientific challenges related to the certification of ML based safety-critical systems, the DEEL (DEpendable & Explainable Learning) project[13], which is born from the international collaboration between the Technological Research Institutes (IRT) Saint Exupéry in Toulouse (France), the Institute for Data Valorisation (IVADO) in Montreal (Canada) and the Consortium for Research and Innovation in Aerospace of Québec (CRIAQ) in Montreal (Canada), aims to develop novel theories, techniques, and tools to help ensure the Robustness of ML based systems (i.e., their efficiency even outside usual conditions of operation), their Interpretability (i.e., making their decisions understandable and explainable), Privacy by Design (i.e., ensuring data privacy and confidentiality during design and operation), and finally, their Certifiability. A white paper [49] released in 2021 introduces the challenges of certification in machine learning, to contribute to this global research effort on the certification of machine learning based safety-critical systems.

3 Methodology

We planned, conducted, and reported our Systematic Literature Review (SLR) process based on the guidelines provided in [122]. In the rest of this section, we will elaborate our search strategy, the process of study selection, quality assessment, and data analysis.

3.1 Search Terms

Our main goal is to identify, review, and summarize the state-of-the-art techniques used to certify machine-learning based applications in safety-critical systems. To achieve this, we select our search terms from the following aspect:

– **Machine learning**: Papers must discuss techniques for machine learning systems. We used the terms *machine learning*, *deep learning*, *neural network*, and *reinforcement learning* to conduct our searches. To include more potential papers, we also used *black box* (deep learning models are often considered as non-interpretable black boxes [31, 205], so certification approaches for such black boxes can be useful), *supervised* and *unsupervised* (because people may use the term "(un)supervised learning" to refer to their machine learning algorithm).
– **Safety-critical**: In this study, we consider that the systems, which need to be certified, are safety-critical, such as self-driving, avionic, or traffic control systems. We use the terms *safety critical* and *safety assurance* to search for this aspect.

[13] `https://www.deel.ai`

```
(
  ("safety critical" OR "safety assurance")
  OR
  (
    (certification OR certified OR certify) AND (automotive OR drive* OR driving OR pilot
         OR aerospace OR avionic)
  )
)
AND
("machine learning" OR "deep learning" OR "neural network*" OR "black box" OR
     "reinforcement learning" OR supervised OR unsupervised)
```

Fig. 1 Search terms used in our study and their logical relationship. The terms in quotes denote the exact match to a phrase.

– **Certification**: *certification* itself is the key term we need to search for. However, through some experiments, we saw that using this term alone will yield a lot of irrelevant results, *e.g.* "Discover learning behavior patterns to predict certification" [266], where certification denotes an educational certificate. We also observed that two major topics of papers can be returned using the terms of *certification*, *certify*, or *certified*, *i.e.* medical related papers and transportation related papers. In this study, we are more particularly interested in the certification of the transport systems, therefore we use the following terms to limit the certification scope: *automotive*, *driv**, *pilot*, *aerospace*, and *avionic*.

  Figure 1 shows all the search terms used and their logical relationship.

3.2 Scope

As the subject of certification in machine learning can be pretty vast, we chose to restrict ourselves to any methods that tackle certification efforts at the algorithm/system structure level of a ML system from the lens of transportation systems. As such, we won't discuss about:

– **Hardware**: If hardware is also one possible source of error for ML applications, we will not consider them in this SLR, as the root cause of such an error is not necessarily the ML system itself.
– **Security/Privacy**: Security threats are an important point of safety-critical ML systems such as an autonomous car for instance. However, such considerations are not limited to ML systems as security is a major issue in all systems. As such, we won't develop on security related problems in ML.
– **Performance only**: As the end goal is to certify the safety of ML system, papers only describing an improvement of performance such as accuracy without any safety guarantees won't be considered in this review.
– **Techniques tied to a specific field** (other than transportation): while we aim for general certification in ML, we mainly focus on the transportation field. If a technique cannot be extended to transportation problems,

for instance because it entirely revolves around a particular field of consideration, we won't consider it.

3.3 Paper search

Inspired by [218, 246], we searched papers from the following academic databases:

- Google Scholar[14]
- Engineering Village (including Compendex)[15]
- Web of Science[16]
- Science Direct[17]
- Scopus[18]
- ACM Digital Library[19]
- IEEE Xplore[20]

Certification of machine learning systems is a relatively new topic. To review the state-of-the-art techniques, we limited the publication date from January 2015 to September 2020 (the month when we started this work) in our searches. As searching key terms throughout a paper may return a lot of noise. For example, an irrelevant paper entitled "Automated Architecture Design for Deep Neural Networks" [1] can be retrieved using the terms mentioned in Section 3.1, because some terms can appear in the content of the paper with other meanings (where the key terms are highlighted in bold): *"... With my signature, I **certify** that this thesis has been written by me using only the in ... **Deep learning** is a subfield of **machine learning** that deals with deep artificial **neural networks** ... in a network to work with different combinations of other hidden units, essentially **driving** the units to"*. The sentence *"I **certify** that this thesis"* appears in the declaration page, which misled our result. Therefore, we restricted our searches only from papers' title, abstract, and keywords. Note that we adapted the pattern depending on the requirements of the database we searched on.

We leveraged the "Remove Duplicates" feature of Engineering Village, by keeping Compendex results over Inspec, to reduce our workload in the step of manual paper selection, described in the next section.

Unlike other academic databases, Google Scholar (GS) does not provide any API or any official way to output the search result. It only allows users to search terms from the paper title or from the full paper. Also, GS does not encourage data mining with it and can at most return 1,000 results per search. To tackle these issues, we:

[14] https://scholar.google.com
[15] https://www.engineeringvillage.com
[16] https://webofknowledge.com
[17] https://www.sciencedirect.com
[18] https://www.scopus.com
[19] https://dl.acm.org
[20] https://ieeexplore.ieee.org

```
I. ("safety critical" OR "safety assurance")
   AND
   ("machine learning" OR "deep learning" OR "neural network*" OR "black box" OR
        "reinforcement learning" OR supervised OR unsupervised)

II. (certification OR certified OR certify)
    AND
    (automotive OR drive* OR driving OR pilot OR aerospace OR avionic)
    AND
    ("machine learning" OR "deep learning" OR "neural network*" OR "black box" OR
         "reinforcement learning" OR supervised OR unsupervised)
```

Fig. 2 Search terms used for Google Scholar. The union of two expressions is equivalent to the one shown in Figure 1.

Table 1 Number of papers retrieved from the academic databases

Database	Number of papers
Google Scholar	11,912
Engineering village	704
Web of Science	146
Science Direct	62
Scopus	195
ACM Digital Library	154
IEEE Xplore	291
Total	13,464

1. Used the "Publish or Perish" tool[21] to automate our search process.
2. Split our search terms into two expressions as shown in Figure 2.
3. Performed searches year by year. Using the above two expressions, we can perform two searches per year, which in turn increases the number of our candidate papers. As we selected to review papers from 2015 to 2020 (six years in total), GS can GS can at most provide us 1,000 $\times$ 6 $\times$ 2 = 12,000 results.

3.4 Paper selection

Table 1 shows the number of results returned from each academic database we used. In the rest of this section, we will describe the steps we used to filter out irrelevant papers and to select our final papers for review.

3.4.1 First Round of Google Scholar Pruning

Google Scholar (GS) returned a lot of results but many of them are irrelevant because GS searches terms from everywhere. We thus performed another round of term search within the results of GS by only considering their paper title and the part of the abstract returned by GS results. At this step, 1,930 papers remained. We further noticed that some surviving results had incomplete or

[21] Harzing, A.W. (2007) Publish or Perish, available from `https://harzing.com/resources/publish-or-perish`

truncated titles, from which we can hardly trace back to the correct papers. Therefore, we manually examined the paper titles and kept 1,763 papers for further pruning.

3.4.2 Pooling and Duplicate Filtering

We pooled together the surviving papers (1,763 papers from GS and 1,551 papers from other databases). Then, we used *EndNote*[22] to filter duplicates. EndNote compares titles and authors to identify potential duplicates. To avoid any incorrect identification, EndNote does not remove duplicates but instead it shows the results to users and asks them to remove duplicates. This mechanism prevents false positives (two different papers can be mistakenly identified as duplicates by the system). There were 727 papers identified as duplicates, from which we manually removed 398 papers. Thus, 2,916 papers (1,741 from GS and 1,173 from other databases) survived at this step.

3.4.3 Second Round of Google Scholar Pruning

To further remove irrelevant results from GS, two of the authors manually examined the title and abstract of the GS papers that survived the previous step. They independently identified any potential papers that are related to the topic of certification of machine learning systems. Knowing that title and abstract alone may not show the full picture of a paper, to avoid missing any valuable paper, the examiners intentionally included all papers in which machine learning techniques are applied to self-driving vehicles, avionic, train, or traffic control systems, because we are especially interested in the certification of transport related systems.

The two examiners then compared their results and resolved every conflicts through meetings until they reached agreement on all surviving papers. As a result, 290 papers survived in this step. Therefore, we obtained a total of 290 + 1,173 = 1,463 papers for further analysis.

3.4.4 Inclusion and Exclusion criteria

To select our final papers for review, two of the authors read title, abstract, and eventually, introduction/conclusion of each surviving paper by considering the following inclusion and exclusion criteria.

<u>Inclusion criteria:</u>

- Techniques that directly strive to certify a machine learning based safety-critical system, *i.e.*, making machine learning models comply with a safety-critical standard.

[22] `https://endnote.com`

– Techniques that indirectly help to certify a machine learning based safety-critical system, *i.e.*, general techniques that improve the robustness, interpretability, or data privacy of such systems.

Exclusion criteria:

– Papers leveraging machine learning to improve other techniques, such as software testing, distracted driving detection. This is on the contrary of our purpose, which is to discover techniques that improve machine learning models' certifiability.
– Papers that thrive to solely improve a prediction metric such as accuracy, precision, or recall.
– Techniques whose scope of application cannot fit into transportation related systems, *i.e.*, we do not consider techniques that are strictly limited to other fields, such as medicine.
– Survey, review papers, research proposals, and workshop or position papers.
– Papers not written in English. Understanding and reviewing papers in other languages might be helpful but is out of the scope of our study.
– Papers which we cannot have access free of charge through our institution subscriptions.

Based on the above inclusion/exclusion criteria, two of the authors independently examined the surviving papers. As a result, 1,000 papers were rejected and 163 papers were accepted by both examiners. We had meetings to resolve the 300 conflicts, among which 86 papers were accepted. Finally, **249 papers were selected for reading and analysis**.

3.5 Quality Control Assessment

The next step is to further control papers' relevance by using quality-control criteria to check that they could potentially answer our research questions, in a clear and scientific manner. As such we devised the following criteria, some of which are inspired from previous systematic literature review studies [55, 246]:

Quality control questions (common to many review papers):

1. Is the objective of the research clearly defined?
2. Is the context of the research clearly defined?
3. Does the study bring value to academia or industry?
4. Are the findings clearly stated and supported by results?
5. Are limitations explicitly mentioned and analyzed?
6. Is the methodology clearly defined and justified?
7. Is the experiment clearly defined and justified?

Quality control questions (particular to our topic):

1. Does the paper propose a direct approach to comply with a certain certification standard?

2a) Does the paper propose a general approach that can help certify a safety-critical system?
2b) If it's a general approach, was it experimented on a transport related system?

We proceeded as before, by evaluating each paper for each criterion, listing them with a score of *0*, *1*, or *2*, with *2* denoting a strong agreement with the statement of the control question. To assess the quality of the papers we read every paper entirely. Each paper was controlled by two different reviewers. For a paper to get validated, each reviewer had to assign a minimum of 7 for the common Quality Control Questions and 1 for the particular Quality Control Questions. If reviewers end up not agreeing, a discussion would ensue to reach an agreement, with a third reviewer deciding if the reviewers couldn't agree after discussion. For those particular Quality Control Questions, note that; first, (1) and (2) are mutually exclusive, so a paper can only score in one of the two and secondly, (2a) and (2b) are complementary, as such their score is averaged to have the result of question (2).

Out of 249 papers, 24 papers were rejected as they were found out after further reading not to comply with the scope and requirements of our SLR. At the time of the release of the preprint, the process is still under construction and the rest of papers are still under cross examination. As such, we for now consider that all papers previously accepted and complying with requirements are selected for the review, that is **225 papers**.

3.5.1 Snowballing

The snowballing process aims to further increase the papers pool by screening relevant references from selected papers. Four interesting papers on ML certification were not discovered by our keywords but were mentioned multiple times in our selected papers as references. We also included these papers in our reading list for analysis.

Hence, in total we collected **229 papers** for examination in our review.

3.6 Data Extraction

For each paper, we extracted the following information:

- Title
- URL to the paper
- Authors
- Year
- Publication venue

Aside from Quality Control Questions, each reviewer was assigned a set of questions to answer based on the paper under review to help the process of data extraction:

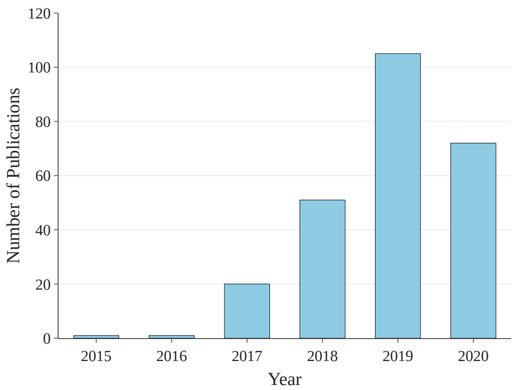

Fig. 3 Number of studied papers per year from 2015 to 2020

- Does the paper aim at directly certify machine learning based safety-critical systems? If yes, what approach(es) is(are) used? Simply describe the idea of the approach(es)
- Does the paper propose a general approach? If yes, what approach(es) is(are) used? Simply describe the idea of the approach(es)
- What are the limitations/weaknesses of the proposed approaches?
- What dataset(s) is(are) used?

Those questions will help us elaborate into the **Section 6**.

3.7 Authors Control

We further asked multiple authors whose paper we considered in our review, and from which we wanted further precision, to give us feedback on the concerned section, in order to make sure we had the right understanding of the developed idea. This allowed us to further improve the resilience of our methodology and review.

4 Statistical results and Taxonomy

4.1 Data Synthesis

From the extracted data, we present statistical description about the pool of papers. Figure 3 shows the number of selected papers published in each year from 2015 to 2020. We observed a general increasing trend of the papers related to the certification topics. The number of papers in 2020 is less than those in 2019 because our paper search was conducted until September 2020. This increasing trend shows that the topic of using ML technique for safety-critical systems attracts more attention over the years. When further looking at Figure 4, we found the topics were well investigated by industrial practitioners

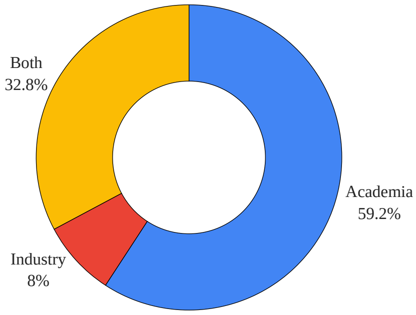

Fig. 4 Authors' Affiliation Distribution

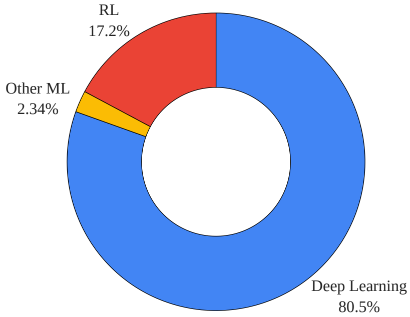

Fig. 5 Categories of the studied papers

or researchers because 41% of the papers were either from companies alone or from a collaboration between universities and companies. This information was deduced by screening papers for author's affiliation and potential explicit industrial grants mentioned.

In terms of studied models in the reviewed papers, Figure 5 shows that 80.5% of the papers employed (deep) neural networks. These models received great attention recently and have been applied successfully to a wide range of problems. Moreover, they benefit from a huge success on some popular classification and regression tasks, such as image recognition, image segmentation, obstacle trajectory prediction or collision avoidance. More than 17% of papers studied RL since it has been well investigated for its transportation-related usage. In particular, researchers intended to apply RL to make real-time safe decisions for autonomous driving vehicles [92, 146, 16].

Figure 6 illustrates the categorization of the studied papers. Robustness and Verification are the most popularly studied problems. Some of the categories can be further split into sub-categories. For example, ML robustness includes the problems of Robust training and Post-training analysis. Similarly, ML verification covers testing techniques and other verification methods *e.g.* formal methods. However, we only see 18 papers (8%) proposing a direct certification technique for ML based safety-critical systems. In other words, although researchers are paying more attention to ML's safety (illustrated by Figure 3), most of our reviewed approaches can only solve a specific problem under a general context (which is on the contrary of meeting all safety requirement of a standard under a particular use scenario, such as autonomous driving or piloting).

Figure 7 shows the papers distribution across venues. If ML related ones are vastly represented with NIPS, CVPR and ICLR being the most represented, papers also come from a wide range of journals/conferences across the board, mainly in computer science/engineering related fields.

Figure 8 represents the distribution of the datasets used across the papers we collected. Note that we only showed the datasets that appeared in more than three papers for readability purposes. Classical computer vision datasets such as MNIST/CIFAR/ImageNet are the most represented, probably because

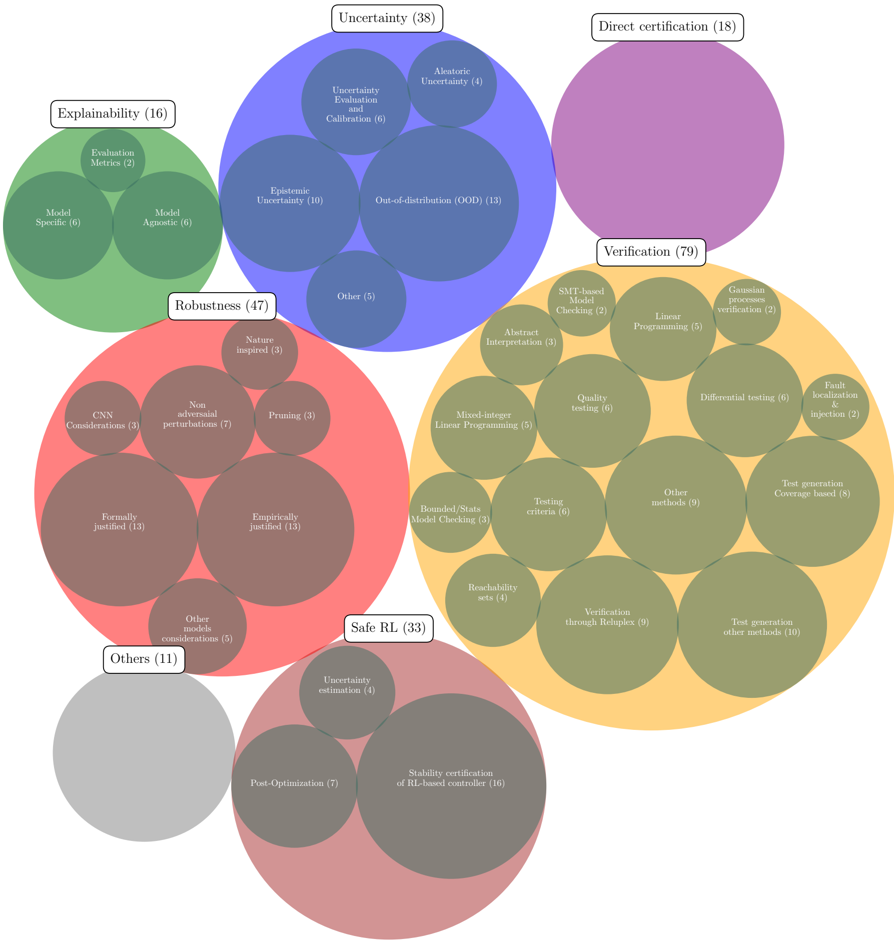

Fig. 6 Categories of the studied papers. Some papers might be present in multiple categories.

they are easy for humans to interpret and straightforward to train/test a model on, which make them perfect candidates for most experimentation. We also note the presence of many driving related datasets such as Cityscape, Traffic Signs, Driving Scenes etc. illustrating the interest of the research toward the development of systems for automotive. In non-computer vision, AXAS-XU is the most represented, especially as it is used in formal verification techniques to assess effectiveness and performance of the technique as the system can process the information from the flight of an unmanned drone, which makes it interesting especially from a safety-critical point of view.

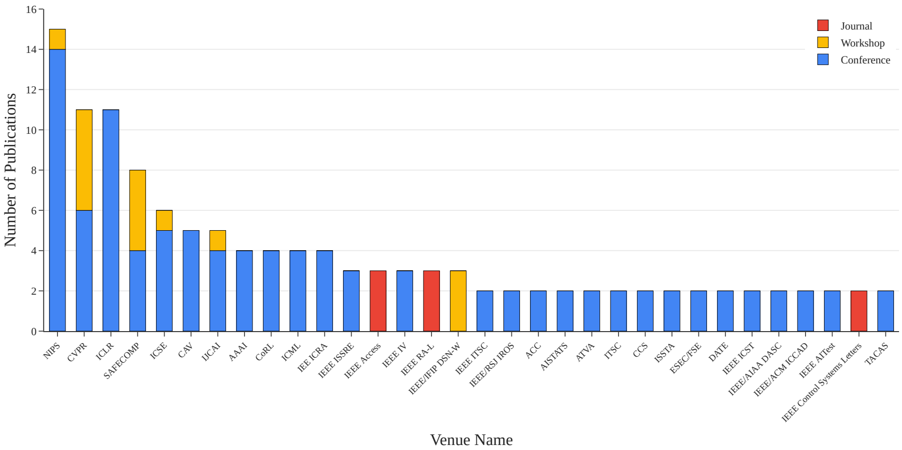

Fig. 7 Papers venue representation in our selected papers. Only venue being represented more than 2 times are represented for readability purpose. 35 papers were found only on Arxiv and are not represented here.

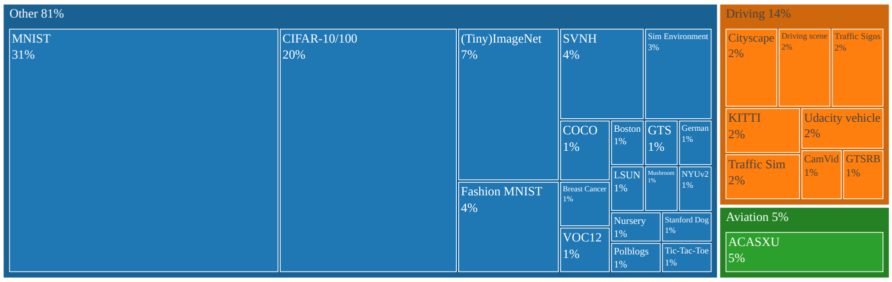

Fig. 8 Dataset repartition by field (Autonomous Driving, Aviation, and Other). The area of a specific rectangle is proportional to the number of distinct papers where the associated dataset was used for experiments. Percentages approximately show the ratio between the area of each rectangle and the whole area, which helps making comparisons between datasets and across fields.

4.2 Selected Taxonomy

We chose to build our review/discussion of the literature around distinct categories of research in ML certification. These categories were extracted through our various fields of expertise and using the DEEL white paper [49] we mentioned earlier as a reference. Hence, the following categories were considered:

- **Robustness**: Robustness deals with methods that aim to improve a model resilience when outside of conditions of operations. In particular, it is the category dealing with Adversarial Examples and how to defend models against them.
- **Uncertainty**: Uncertainty deals with the ability of a model to acknowledge its own limitations and with techniques related to the calibration of the

margin of error a model can make because of its specificity and data used. It also covers the concept of Out-of-distribution (OOD), that is inputs that are, in some sense, too far from the training data.
- **Explainability**: Explainability is about all techniques that can shed some light on the decision a model makes, that is to remove the traditional "black-box" property of some models.
- **Verification**: Verification encompasses both Formal/Non-formal methods of verification, that is methods that aim to prove mathematically the safety of a model in an operation range, as well as all Testing techniques that are more empirical and based on criteria of evaluations.
- **Safe Reinforcement Learning**: By its very nature, RL differs from supervised/unsupervised learning and, as such, deserves to have its own dedicated category. This covers all techniques that are specifically tailored to improve the resilience and safety of a RL agent.
- **Direct Certification**: Direction Certification contrasts sharply with previous categories; where others categories are concerned with existing techniques tackling one or two aspects of certifiability, this category deals with higher-level methods that discuss the possibility of obtaining a full certification process for ML models, especially by taking into accounts all the points raised by the others categories
- **Others**: This category includes papers that couldn't fit in the other categories, in particular because they lie at the crossroads of several categories or raise a potential new field for certification.

Section 5 will elaborate on the different categories, and for each individual one, we shall present the state-of-the-art techniques gathered from our SLR, explain their core principles, discuss their advantages/disadvantages, and shed light on how they differ from one another. Figure 9 provides an overview of how the main categories will be tackled.

Afterward, **Section 6** will develop on the global insight extracted from the SLR. We shall highlight the existing gaps that could serve as a starting point for future work in the research community.

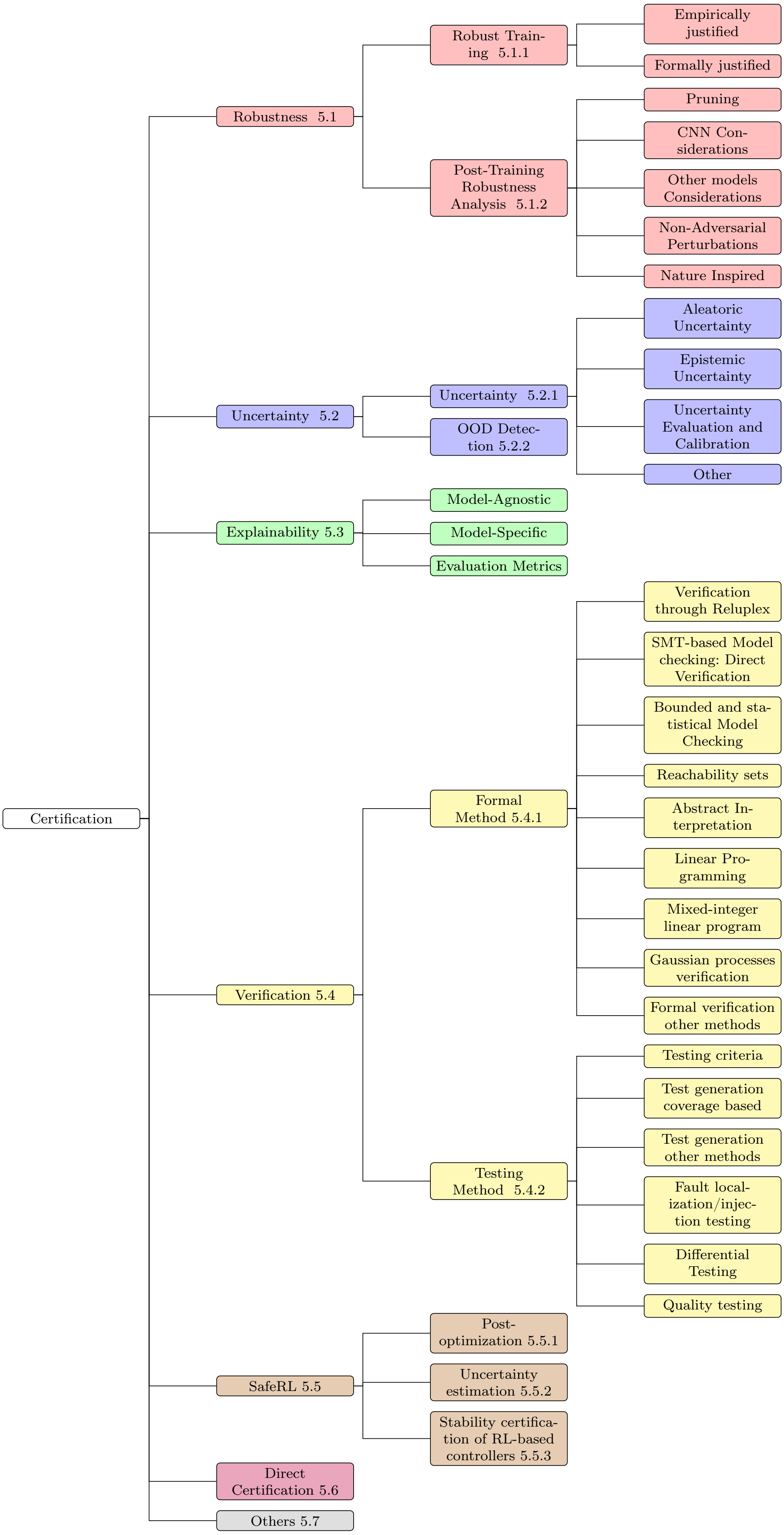

Fig. 9 Overview of techniques discussed in the paper. You can click on one to jump immediately to it.

5 Review

In this section, we summarize the state-of-the-art techniques that can be used for ML certification. Each of the categories is handled separately and hence do not need to be read in a specific order.

5.1 Robustness

Robustness is the property of a system to be more resilient to unknown inputs. Under the ML context, a robust model retains its effectiveness even outside its usual conditions of operation. Today, although ML models have been able to achieve or surpass human performance on certain tasks such as image recognition, they are facing a lot of robustness challenges, which hampers their integration in systems that must satisfy safety standards. In most cases, robustness issues arise because of the *distributional shift* problem, *i.e.* when the training distribution the model was trained on is different from the testing distribution. The most notorious examples of such phenomena are the *adversarial attacks*, where carefully crafted perturbations can deceive a ML model [76]. Formally, given a sound input x, an adversarial example is defined as a crafted input $\tilde{x}$ such as $\|x - \tilde{x}\| \leq \delta$ and $f(x) \neq f(\tilde{x})$ where $f(x)$ is the ML model prediction for a given input and δ is small enough and quantifies the neighborhood where x and $\tilde{x}$ are indistinguishable from one another. One of the first techniques used to generate Adversarial Examples is called the Fast-Gradient Sign Method (FGSM) [76] and is defined as:

$$\tilde{x} = x + \epsilon \times \text{sign}(\, \nabla_x \mathcal{L}(\theta, x, y)\,), \tag{1}$$

where $\mathcal{L}$ is the loss function, θ are the weights of the model and ϵ is a perturbation magnitude, small enough for it to be barely distinguishable from a human point of view, but strong enough to fool a model. This attack slightly perturbs the input so as to increase the loss of the model by following the gradient. Rapidly, it was clear it would be possible to exploit those adversarial examples to strengthen the robustness of a model, by trying to include the adversarial process in the training of the model. This method is referred to as adversarial training. Later, [151] introduced Projected Gradient Descent (PGD) as an improvement over FGSM through multiple iterations of FGSM with random initializations inside a L_∞ ball around x. More complex attacks were later developed such as the C&W attack [30], which is a targeted attack meaning that it selects the specific class the model is fooled towards. As such, this whole problem has raised much attention [184] as it shows the existence of countless inputs on which ML models make wrong predictions, but on which we would expect them to perform accurately.

Adversarial examples aren't the only examples of distributional shift; random perturbations/transformations, without any crafted behaviour, such as Gaussian noises or Out-of-distribution (OOD) examples (see **Section 5.2.2**) can also lead to such a problem. The rest of this section shall elaborate on recent techniques used to increase the robustness of ML systems.

5.1.1 Robust Training

Robust training refers to groups of techniques that aim at increasing robustness to adversarial attacks or distributional shifts by introducing new objectives and/or processes during model training. When only considering adversarial perturbations of the input, the terminology becomes more specific: adversarial training. Some of the recent techniques in the literature were found to be justified purely with empirical evidence while others were derived from formal robustness guarantees. We now discuss methods following these two schools of thoughts separately.

Empirically Justified

[151] were the first to formalize adversarial training as a saddle point optimization problem

$$\min_{\theta} \rho(\theta), \text{ where } \rho(\theta) = \mathbb{E}_{(x,y)\sim D}\left[\max_{\delta \in S} \mathcal{L}(\theta, x+\delta, y)\right], \tag{2}$$

where the inner maximisation problem searches for the most adversarial perturbation of a given data point (x, y). The outer minimization problem attempts to fit the model such that it attributes a low loss on the most adversarial examples. The authors provide empirical evidence that the saddle point problem can be efficiently solved with gradient descent using the following procedure: generate adversarial examples (with FGSM or PGD) during the training phase and encourage the model to classify them correctly. Two recent modifications of this training procedure add a new term to the training objective that encourage adversarial and clean examples to have similar logits [214] or similar normalized activations per layer [136]. Both modifications led to an increase of robustness to FGSM and PGD attacks.

Autoencoders are powerful unsupervised models that compute projections of the data to low-dimensional manifolds. The main idea behind Deep Denoising Sparse Autoencoder (DDSA) is to train an autoencoder to learn the low-dimensional manifold onto which the training data concentrates [17]. Inputs are then passed through the DDSA before being given to the model. The main assumption of this defense is that adversarial perturbations deviate from the training data manifold and are cancelled by projecting back onto it. Empirical evidence shows that this approach alone can significantly improve a model's robustness against FGSM and PGD attacks, but only has a weak effect against C&W attack.

An alternative robust training procedure uses a redundant Teacher-Student framework consisting of three networks named the static teacher, the static student, and the adaptive student [19, 20]. The two students apply model distillation of the teacher by learning to predict its output, while having a considerably simpler architecture. Contrary to both static networks, the adaptive student is trained online and is encouraged to have different layer-wise features from the static student through an additional loss-term called the inverse feature mapper. The role of the adaptive student is to act as a watchdog to point out which neural network is being attacked online through a threshold computation. Similar ideas are used in [217].

Other methods that increase robustness to attacks include adding noisy layers [130, 178], GAN-like (Generative Adversarial Networks) procedures [53, 143], adding an "abstain" option [127], and using Abstract Interpretations, *i.e.* approximating an infinite set of behaviours with a finite representation [160].

Beyond adversarial attacks, distributional shift can also refer to changes in the data-generating distribution over time. When the shift only affects the conditional probabilities of the target given the input *e.g.* $p_{t1}(y|x) \neq p_{t2}(y|x)$, it is referred to as "real concept drift". In such instances, it is necessary to adapt the model architecture over time, which is possible for K-Nearest-Neighbors classifiers and Random Forests as done in [78].

Formally Justified

Now, we describe training procedures that were conceived by first deriving a formal robustness guarantee, and modifying the standard training process so that the guarantee is met. A common theoretical guarantee for robustness is the Lipschitz continuity of the neural network. A network is called Lipschitz continuous if there exists a constant $L \geq 0$ such that

$$\|x - z\| \leq \epsilon \implies |f(x) - f(z)| \leq L\epsilon, \tag{3}$$

which guarantees that, if the inputs is slightly perturbed by a distance ϵ, the output of the network cannot vary by more than $L\epsilon$, guaranteeing robustness. The lowest value of the constant L for which Equation 3 holds is called the Lipschitz constant of the function and is denoted L^*. The exact calculation of this constant is very hard for neural networks and one usually resorts to computing upper bounds on L^*. Recently, tight bounds on the Lipschitz constant of neural networks were obtained by solving a Semi-Definite Problem and including an Linear Matrix Inequality (LMI) constraint during training, a Lipschitz bound can be enforced or minimized [170]. However, the bottleneck of the proposed approach is the Semi-Definite Problem, which has a cubic complexity in terms of the number of parameters in the model.

Another robustness guarantee of neural networks is based on a geometrical argument specific to ReLU activation functions. For such networks, it was proven that the output is linear by part (linear decision boundary for classification tasks) on a set of polytopes that cover the whole input space. A recent theorem guarantees that if a L_1 ball and a L_∞ ball around an instance x are contained in a single polytope and are located far away from the linear decision boundary inside said polytope, then a lower bound on the L_p norm of any effective attack can be obtained [45, 44]. The theorem leads to a novel regularization term to the loss function that pushes the decision and polytope boundaries far away from the training inputs. However, this method increases the test error (robustness vs accuracy trade-off).

An alternative theoretical result provides a lower bound on the minimal L_p perturbations required to change predictions of a continuously differentiable classifier, which includes neural networks with most activation functions excluding ReLUs [94]. Increasing the lower bound, which is done by increasing the difference between the logits of the predicted class and other classes as well as decreasing the difference between the logit gradients, increases the robustness of the network. The authors argue that using a cross-entropy loss already increases the difference between the logits of the predicted class and other classes, and therefore they introduce a novel regularization term called the Cross-Lipschitz Regularizer that encourages small differences between logits gradients, indirectly maximizing the theoretical bound.

When considering robustness to distributional shift instead of only adversarial examples, a common training approach is Wasserstein Distributionally Robust Learning (WDRL). The general principle of WDRL is to consider the worst possible generalisation performance of the model under all data-generating distributions which are "close" to the original distribution used for training. In this context, closeness is measured in terms of the so-called Wasserstein Distance [207]. Although the Wasserstein distance is a very powerful mathematical tool, the formulation of WDRL is intractable and must be approximated/reformulated for specific applications. For example, in the context of Model Predictive Control (MPC), WDRL is reformulated using a convexity-preserving approximation with minimal additional computation [116]. For general supervised learning applications, [207] relaxed WDLR into a Lagrangian form optimizable with stochastic gradient methods. More recently, it was proposed to modify the Lagrangian form to consider heterogeneity of the data e.g. the fact that some features are more uninformative/noisy than others [142]. The main principle of the approach is to favor larger distributional shifts along directions associated with noisy features, thus breaking spurious correlations.

PAC-Bayes theory provides upper bounds on the expected performance of a learned distribution on models. While only a finite amount of data is available for training, this theoretical approach allows to calculate guarantees (holding with high probability) on the average performance of the model when deployed on arbitrary samples coming from the distribution from which the training set is assumed to be sampled in an i.i.d fashion. In this sense, PAC-Bayes theory aims at learning a distribution of models from a finite amount of data with a guarantee on the generalization risk, which can be viewed as a form of robustness. Gaussian Processes constitute a popular class of distributions on models where PAC-Bayes bounds can be evaluated analytically and even minimized, leading to novel robust training procedures [180].

The robustness of Deep Reinforcement Learning to small perturbations of the observed states (induced by sensor noise or adversarial attacks) has recently been tackled. To attain robustness, [57] propose using novel robust Q-values which are the lower bound on all possible Q-values when considering the set of possible deviations from the observed state, which is modeled with an ϵ-ball in state space. Therefore, using these Q-values, agents do not simply take actions based on the state they are in, but also based on all small perturbations of this state.

In the context of dynamic systems modeling, the term robustness is used interchangeably with the term stability, which refers to the property that the modeled system should not diverge as time updates are applied. Recent ML-based modelings of dynamical systems with formal robustness guarantees include Lyapunov networks [188], a convex reparametrization of Recursive Neural Networks [185], and linear dynamical systems with partial state information extracted from images [48]. A notion very similar to stability of dynamical modeling is the "temporal consistency" of semantic segmentation of videos, *e.g.* the constraint that an object should not appear/disappear in consecutive frames [232]. Combining the notions of stability and temporal consistency, we can state more generally that systems evolving over time are robust if they behave appropriately at any time t. This definition simultaneously includes behavior over infinitesimal time steps and behavior as time goes to infinity.

5.1.2 Post-Training Robustness Analysis

This subsection encompasses considerations and analysis made post-training, in order to study extensively robustness behavior of the model and how it can be further improved.

Pruning methods

Robustness improvement can also be obtained thanks to post training analysis, by reusing this information in order to improve the model resilience. Pruning techniques, which aim to remove weights connections from pre-trained neural networks, are such methods. While pruning was used before mainly in order to reduce the size of a model so it could fit on smaller systems, recent studies have shown it can surprisingly be used to improve robustness. Pruning is generally done following certain metrics, generally the weights with the lowest magnitude are pruned, however [201] showed that its effectiveness against adversarial attacks is reduced and it is therefore proposed to use pruning based on importance score that are scaled proportionally to pre-trained weights to fasten computation. An optimization step can then be realized in order to further prune the model. They showed that the reduced model is almost as robust as the original model. A central point in pruning is the "Lottery Ticket Hypothesis", which roughly refers to the idea that any randomly initialized neural network contains a sub network that can match the original network accuracy when trained in isolation. [43] tested this hypothesis and confirms that the pruned model tends to be more robust to adversarial attacks while being faster to be trained. [257] is an example of pruning improving both robustness and compression. Aside from the trade-off problem between keeping accuracy and reducing size, pruning methods generally suffer from less optimized architecture as weights are removed disorderly across the model.

Considerations in Convolutional Neural Networks

Convolutional Neural Networks (CNN) were studied extensively as they are the base for a lot of applications; [11] studied adversarial robustness in Convolutional Neural Networks, showing that residual models such as ResNet are more resilient than chained model such as VGG. They also pointed out that multiscale processing makes the model more robust. [264] observed that neurons of a convolutional network activated by normal examples follow the rule of "vital few and trivial many". In other words, in a convolutional layer, only a few neurons (related to useful features) should be activated and many other ones (related to adversarial pertubations) should not. Hence, they proposed an idea to select the vital few neurons that can retain the model accuracy equivalent to the original model. The neuron selection starts backward from the last layer but skips the first layers because the authors did not find vital neurons in those layers. [144] also discovered the differences between clean and adversarial examples in terms of neuron activation and data execution paths. They introduced a tool to generate data path graphs, which visually shows how a pre-trained CNN model processes input data. The tool takes feature maps as the basic unit in the graphs. Machine learning engineers can compare the data paths between adversarial images and their original images to analyze at which layer the ML model diverges the decision and use their tool to diagnose how a CNN model fails to classify adversarial examples. Note that this technique, while focusing on robustness, also fills the role of explainability technique touched on in **Section 5.3**.

Other models considerations

As for other models, [243] studied k-nearest neighbors robustness, showing that the value of k as well as the number of data points both influence robustness, focusing more on a robust version of the 1-NN. Ensemble learning (that is, training multiple models independently on the same datasets and averaging predictions to have a global one), is known to be more resilient than a single model [80], as redundancy can provide extra security. [153] pushes the step further, by training each model of an ensemble to be resilient to a different adversarial attack by injecting a small subset of adversarial examples, which profit to the ensemble globally, even though it comes at the cost of training more models. Finally, [112] showed on Graph Neural Networks (GNN), that it is possible to use graph properties such as low-rank and sparsity to increase the model robustness as adversarial examples don't follow such properties. If such an example is only applicable to GNN, it shows that the identification of relevant mathematical properties of a model can help to improve robustness. As a last note, class distribution itself can also be a vector of robustness. Indeed, [168] showed that some classes can be more likely to flip to another when under adversarial perturbation. By analyzing the nearest neighbors map, it's possible to identify such classes and to increase the number of examples belonging to those to retrain the model. In a sense, it shows that class unbalance can also affect not only the model predictions, but also its robustness against adversarial examples.

Non-adversarial Perturbations

Distributional shifts were also studied more empirically from the point of view of image transformations and noise injection. [97] designed an ImageNet benchmark by adding corruption and/or perturbations in order to evaluate the resistance to corrupted images of different models. Those images aren't adversarial, as they aren't specifically designed to make the model fail, but are general transformations that cause a shift in the prediction of the model while being relatively close to the normal images. They notably show empirically that even small non-adversarial perturbations can lead a model to error. [10][163] showed similar observations and indicate that using such images as data augmentation can boost robustness against adversarial examples. Note that [163] used synthetic transformation of ADAS based images, evaluated through robustness landscape, which shows such a concept can be applied in transportation related settings. In fact, similar perturbations were shown to be useful in defending against adversarial examples [42], by pre-processing potentially adversarial examples with those transformations or by using adversarial re-training [111]. In particular, [111] used noise injection technique to strengthen model's robustness; the network is updated in the presence of feature perturbation injection to improve adversarial robustness while the parameters of the perturbation injection modules are updated to strengthen perturbation capabilities against the improved network. The idea behind this is that, when both the network parameters and the perturbed data are optimized, it is harder to craft successful adversarial attacks and it seems empirically to be more effective than some adversarial training methods such as [178][130].

Nature inspired techniques

Although these techniques also apply to the training stage, they differ from the aforementioned ones in that they tackle the adversarial attacks from a completely different angle. [47] observed that CNN models with hidden layers are somehow closer to the primate primary visual cortex (V1) and are more robust to adversarial attacks. This inspiration leads to a novel CNN architecture where the early layers simulate primate V1 (the VOneBlock), followed by a neural network back-end adapted from the existing CNN models (*e.g.* ResNet). The paper showed a number of interesting findings such as evidence that the new architecture can improve the robustness against white-box attacks. It is worth noting that this approach can outperform other defense based on adversarial training such as [193] (L_∞ constraint and adversarial noise with Stylized ImageNet training) when considering a wide range of attack constraints and common image corruptions. Similarly [259], inspired by the association and attention mechanisms of the human brain, introduced a "caliber" module on the side of a neural network to replicate such a mechanism. Traditionally, we assume the training data distribution and operation data are independently/identically distributed which is not always the case in practice (*e.g.* image with sun in training vs rainy day in operation), which leads to retraining and/or fine-tuning in order to account for that, which can potentially be potentially very expensive. They instead propose to retrain only the light weight caliber module on those new data, to help the model "understand" those. In practice, they minimize the loss between output/target, except that the output is computed by point-wise multiplication of input with a perturbation that is the output of the caliber module.

- Robust (Adversarial) training encapsulates all methods that modify the training procedure of a model in order to increase its robustness.
- There exists a wide array of robust training procedures that are based on formal theoretical guarantees, which should be investigated extensively.
- Post-training empirical observations or model considerations can also serve as a base for Robustness improvement, with for instance analysis of non-adversarial perturbations or interactions of neurons inside of a model.
- However, a more thorough understanding of how such observations can indeed strengthen a model's robustness is needed for those methods to be effectively applied to certification of safety-critical systems.

5.2 Uncertainty estimation and OOD detection

The concepts of uncertainty and OOD detection are closely related. In fact, the former is often used as a proxy for the latter. For this reason, this section starts by delving deeply into uncertainty quantification, before discussing the topic of OOD detection.

5.2.1 Uncertainty

In Machine Learning, uncertainty generally refers to the lack of knowledge about a given state, and can be categorised as either aleatoric or epistemic [119]. On the one hand, aleatoric uncertainty measures the stochasticity that is inherent to the data, and can be induced by noisy sensors or a lack of meaningful features. On the other hand, epistemic uncertainty refers to the under-specification of the model given the finite amount of data used in the ML pipeline. This uncertainty is assumed to be reducible as more and more data is available, while aleatoric uncertainty is irreducible.

Aleatoric Uncertainty

The aleatoric uncertainty quantifies the amount of noise in the data and is formalized by treating both inputs x and output y as random variables which follow a joint probability distribution $(x, y) \sim D$. The stochasticity of the distribution D is inherent to the task at hand and cannot be reduced by considering larger datasets.

In a regression setting, the most common technique to estimate the aleatoric uncertainty of the data is to fit a neural-network using the loss attenuation objective [119]. The main idea behind this objective function is to assume a Gaussian conditional distribution of y given the input x *i.e.*

$$y \mid x \sim \mathcal{N}(f_\theta(x), \sigma^2_\theta(x)), \tag{4}$$

where $f_\theta(x)$ and $\sigma^2_\theta(x)$ are both outputs of the network, and represent the prediction, and the aleatoric uncertainty respectively. The choice of Gaussian conditionals is often made for computational convenience but can also be justified by the central limit theorem. Now, the loss attenuation objective is defined as the logarithm of the conditional distribution, averaged across the training set

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim D}\left[\frac{(y - f_\theta(x))^2}{2\sigma^2_\theta(x)} + \log \sigma_\theta(x)\right]. \tag{5}$$

Although this loss function is derived from a regression setting, we have observed some efforts to adapt it to other tasks such as classification [129], and space embeddings for visual retrieval systems [219].

Loss attenuation has the drawback of requiring a modification of the training objective as well as the network architecture since the network must output both the mean and variance of the conditional distribution. A model-agnostic alternative to estimate the aleatoric uncertainty is to propagate the sensor noise through-out the network using a technique called Assumed Density Filtering (ADF) [199]. The main assumption is that the target has a deterministic relationship with a noisy perturbation of the input *i.e.* $y = f_\theta(z)$ where $z \mid x \sim \mathcal{N}(x, \Sigma_{\text{sensor}})$. The covariance Σ_{sensor} is dependent on the sensors and must be carefully calibrated.

Epistemic Uncertainty

Epistemic uncertainty encapsulates our lack of knowledge about the task induced by the finite amount of data used in the ML pipeline. Because of this lack of data models are under-specified, meaning that there exists a wide diversity of models that perform equivalently well on any given task. The quintessential technique used to estimate this type of uncertainty in deep learning are Bayesian Neural Networks (BNN), which work with a probability distribution over good parameters called the posterior distribution instead of working with a single set of optimal parameters. The posterior distribution of BNN provides a way to formalize epistemic uncertainty which yields predictive uncertainty through integration. These integrals in high dimensions can be effectively estimated using a Monte-Carlo approach in the case where samples from the posterior distribution are available. However, for most neural-network architectures, the true posterior distribution over parameters is intractable and must be approximated. The most common approximation scheme seen in the literature is to use dropout at test time also called MCDropout [119, 204], which was successfully applied to automonous driving [133], robot control [224], optical character verification [186], and health pronostics [175]. Given a specific input x, the variance of the model's predictions from different forward passes with random shutdown of neural activations is used as a measure of the epistemic uncertainty at x. The principle cost of this approach comes from the several forward-passes required to estimate the epistemic uncertainty at a single input. However, some efforts are being done to estimate the variance of the network's output with a single forward-pass, using first-degree Taylor expansions of the network layers [176].

An alternative to MCDropout seen in the literature suggests to approximate the posterior by collecting parameter samples using Stochastic Gradient Descent: every time the optimizer gets stuck in a local optima, current parameters are saved before Gaussian noise is injected to escape the local optima and the process is repeated [169]. Deep Gaussian Processes were also recently proposed as a non-parametric alternative to BNNs which do not require a sampling procedure to provide epistemic uncertainty measures, although it is not yet clear how well they scale considering they were only tested on MNIST and Rectangles [110].

A very promising alternative to BNNs has recently been developped: the so-called *Deep Evidential Regression* [6], and is based on computing a distribution over likelihoods allowing for the simultaneous estimation of aleatoric and epistemic uncertainties without need of Monte-Carlo sampling. This method was shown to be competitive with the state-of-the-art while being three times faster, as it requires only a single forward pass through the network.

Uncertainty Evaluation and Calibration

One of the main challenges in uncertainty quantification is the absence of a ground truth for the exact values uncertainties should take. Indeed, unless one is working on toy-datasets on which the data-generating distributions D are explicitly known, there is no universal way to compare uncertainty estimations. A solution to this issue is to evaluate uncertainties based on their usefulness for increasing model safety at prediction-time.

Generally, uncertainties are used as measures of confidence that the model has in any given prediction. In safety critical systems, confidence measures are useful as accurate proxies for prediction error *e.g.* the confidence should be high for correct classifications and low for incorrect ones. This is such an important property that in some work, uncertainty is directly defined in terms of the ability to predict erroneous decisions [123]. One proposed uncertainty evaluation that goes along these lines are the Remaining Error Rate (RER) and the Remaining Accuracy Rate (RAR), which are the ratio of all confident miss-classifications to all samples and confident correct-classifications to all samples respectively. A ROC-like curve of the two quantities for different confidence thresholds can then help comparing different uncertainties estimators [100]. More specific uncertainty evaluations can be used depending on the task. Notably, in autonomous driving control, the uncertainty can be evaluated as a proxy of the risk of crashing in the next n seconds [159].

Moreover, it is important to ensure that uncertainty estimations are calibrated. For example, in the context of classification, uncertainties are usually within the $[0, 1]$ interval which makes it tempting to interpret them as probabilities. However, these "probabilities" do not necessarily have a frequentist interpretation, *e.g.* it is not because a system makes 100 predictions each with 0.9 certainty that about 90 of those predictions will indeed be correct. When the uncertainties of classifications are close to their frequency of errors, the uncertainty method is said to be well-calibrated.

Although the concept of calibration is intuitive for classification, it is far less straight-forward to define in regression. This difficulty comes from the fact that uncertainty estimators yield a probability density function of the target given the input, see Equation 4. What is the frequentist way to interpret this distribution? In the literature, a first way to assert calibration is that quantiles of the conditional distribution of y given x should approximately match their empirical estimates on the whole dataset [59]. This formulation has however recently been criticized and redefined in terms of the ability of the uncertainty estimator to predict the Mean Square Error of the model at any datapoint [134].

Common techniques for calibration such as isotonic regression and temperature scaling are post-hoc meaning they can modify any uncertainty estimation after the model is trained. Said approaches however require a held-out dataset called the calibration set in order to be applied. It is also possible to calibrate uncertainties by modifying the training objective so it directly takes calibration into account [59].

We observe that there is a growing literature on uncertainty calibration of neural-networks-based object detectors [59, 134, 124]. Uncertainties of object detection are especially challenging to calibrate because these models involve both an object classifier and a bounding box regressor, whose uncertainties must be simultaneously calibrated.

Other

The way uncertainty was previously defined mostly applies to supervised learning tasks. Indeed, uncertainty measurements were used in tandem with model predictions to provide proxies of prediction confidence. However, some papers in the literature were found to use uncertainty in a more general sense. For example, [231] studies the optimization of convex objective functions under uncertain constraints. Uncertainty in this context refers to the lack of knowledge about the constraints of the task.

Moreover, we found multiple studies where uncertainty estimations were applied to Markov Decision Processes [228], and dynamical systems [61, 58], allowing for safer autonomous control. Note that the definition of "safe" is application dependent and must be derived from domain knowledge. For example, in autonomous driving, criteria for safe control could be to have a low path curvature and being far away from lateral obstacles [261]. On these tasks, uncertainty measurements are not really used as proxies for prediction confidence, but more as tools that allow autonomous agents to dynamically update the amount of information they know about their environment.

5.2.2 OOD detection

Out-of-distribution (OOD) refers to input values that differ drastically from the data used in the ML-pipeline for both training and validation. Unsafe predictions on these OOD inputs can easily occur when a model is put "in-the-wild". This unwanted behavior of a ML model can be explained on the one hand, by the lack of control on the model performance on data that differ from the training/testing data, *i.e.* OOD data; and on the other hand, by the difficulty to specify which inputs are safe for the trained model, *i.e.* in-distribution data. Traditionally, OOD instances are obtained by sampling data from a dataset that differs from the training one. In the context of certification for safety-critical systems, it is primordial to either provide theoretical guarantees on how models would perform in production (robustness), or develop techniques to detect OOD inputs, so that the prediction made by the model can be safely ignored or replaced by decisions made by a human in-the-loop. This subsection focuses on various techniques used to detect OOD inputs.

For classification, the simplest approach to this detection task is to rely on the maximum softmax probability of the last layer of the network [98] as a proxy for the uncertainty on the prediction. If this method sometimes succeeds in detecting OOD datasets sufficiently different from the in-distribution, it's not necessarily the case for more closely related datasets. Moreover, softmax was shown not to be a good measure of model confidence [66, 98], as it can yield high value for data far from training distribution. Other methods were proposed to tackle this issue, the two most well-known being ODIN [139] and Mahalanobis Distance [131]. While ODIN uses temperature scaling to calculate a better softmax score, Mahalanobis Distance based detection makes the assumption that, in each layer, the features of each class follow a Gaussian distribution. Each Gaussian is estimated by a class mean vector together with a layer specific covariance matrix shared by all classes. The final score is obtained as a weighted sum of the Mahalanobis distance in each layer, where the weights are tuned using OOD samples. One thing those two methods have in common is that they both pre-process the inputs by adding some gradient-based perturbation. The idea is similar in spirit to that of adversarial attacks, but here the perturbation is calculated so as to increase (instead of decrease) the confidence score of the predicted class, with the goal that this will have a stronger effect on in-distribution examples, making them more separable from out-of-distribution ones. In both cases, a decision threshold between OOD or in-distribution must then be fine tuned on some validation OOD dataset.

The idea of using input perturbations to better distinguish OOD from in-distribution have been echoed in many methods; [98] used noises as a way to train a classifier to distinguish noisy or not images based on a confidence score. [183] used them to disentangle what they call background (population statistics) and semantic (in distribution patterns) components which are two parts of the traditional likelihood. They argue that the background part is why OOD can be misinterpreted. Indeed, they observed both that several OOD can have similar background components as in-distribution data and that the background term can dominate the semantic term in the likelihood computation. By adding noise, they essentially mask the semantic term so they can train a model specifically on background components. This could explain why models such as PixelCNN can fail on OOD detection.

Using the softmax as a proxy for OOD detection is known to yield overly-confident predictions on OOD samples. In fact, this has recently been mathematically proven for ReLU networks [95]. Therefore, recent methods attemps to find other proxies for OOD detection; GLOD[23] [7] replaced this softmax layer with a Gaussian likelihood layer to fit multivariate Gaussians on hidden representations of the inputs. These class-specific probability density functions are used to compute the log-likelihood ratio between the class predicted and other classes, with a threshold to detect OOD. This method does not use an OOD dataset for fine tuning but its performance is lower in terms of model accuracy and detection rate compared to state-of-the-art models. [99] tackled the problem of OOD detection for multi-class by using the negative of the max unnormalized logit, since traditional maximum softmax OOD detection techniques fail as the probability mass is spread among several classes. [242] used a generative model for each class of a dataset. This way, they can use those models at test time to see from which instance an input is closest, and compare it to a threshold tuned on in-distribution data. GraN [145] assumed the parameter gradients of the loss computed after forward propagating x can be used to distinguish miss-classified examples as they should have larger gradients, which make using a simple trained logistic regression enough to detect them. Similarly, autoencoders can also be used [84] as a way to recognize activation patterns of correctly classified inputs in order to detect at inference OOD through pattern detection.

An alternative to using proxies for OOD detections is to use a purely data-modeling approach such as the one in [156], where the full joint distribution of x and y is estimated with Gaussian Mixture modelings of in-distribution and out-of-distribution images. The probability density for OOD images is calibrated on the 80 Million Tiny Images dataset, which is used as a proxy for all possible images. It is also possible to detect OOD by partitioning the input space with a decision tree, identify the leaves with low training point density, and reject any new instance that lands in those leaves [85].

OOD detectors must be evaluated on their ability to reject OOD inputs without rejecting too many in-distribution inputs, which would make the ML system too passive. [102] propose a set of evaluations for OOD detection, using traditional metrics such as Area Under Receiver Operating Characteristic (AUROC), and new ones such as Coverage Breakpoint at Full Anomaly Detection (CBFAD) which quantifies how many in-distribution are left out for a

23 Author's remark: A new improvement of GLOD, FOOD, was released earlier this year. Following our methodology, we kept only GLOD reference, but we invite reader to check the new instalment of the method: `https://arxiv.org/abs/2008.06856`

threshold that allows to detect all OOD. It was observed that, in general, OOD detection methods suffer from too many False Positives depending on the setting and the task they are applied to. Moreover, such detectors can also show lower performance on Adversarial Examples, or even on Adversarial OOD (*i.e.* OOD to which we applied adversarial perturbations). [200] analyzed such a problem, showing that neither OOD nor Adversarial example detector is good against adversarial OOD. They recommend adding a small subset of OOD in the adversarial training to boost defense. However, little work has been made to verify methods in real safety-critical settings, which are the settings in which OOD detection is the most pertinent and important. It's crucial that such methods are generalizable to account for all potential OOD samples the system might come across with; in general, authors do not seem to consider that OOD samples follow a distribution (except for [156]), but instead seem to use this term informally to refer to "any sample that is different enough from the training set or distribution". In this context, the performance of a method depends heavily on the particular choice of OOD samples used for the evaluation and these scores must therefore be considered with caution. Moreover this performance comes with no theoretical guarantees, especially when it comes to the frequency of False Negatives, *i.e.*OOD samples which are detected as safe for prediction, which is critical for certification.

- There is no universal metric to evaluate uncertainty so different uncertainty estimators must be compared on their usefulness in increasing the safety of ML systems by acting as proxies for prediction error, and by having frequentist interpretations on in-distribution data *e.g.* being calibrated.
- OOD detection is generally based on finding a mean to disentangle in from out distribution data, then applying a threshold cut that is correctly tuned. Although the state-of-the-art provides good results on traditional datasets/models, getting a better understanding of how neural networks inner representations differ from in/out of distribution is necessary to improve on existing methods.
- Moreover, most techniques, while being pretty efficient, are evaluated on a limited number of OOD datasets. As such, there is at best only strong empirical evidence a technique can work and no guarantee it can generalize to every possible OOD the model might come across, which is not sufficient for safety-critical application. A better understanding of what constitutes all of OOD for a model (and a given task) could lead to formal guarantees such as it is the case for Adversarial Examples.

5.3 Explainability

With the steady increase of complexity of machine learning models and their wide-spread use, a growing concern has emerged on the interpretability of their decisions. Such apprehensions are especially present in contexts where models have a direct impact on human beings. To this end, the European Union has adopted in 2016 a set of regulations on the application of machine learning models, notably the "right to explanation", which forces any model that directly impact humans to provide meaningful information about the logic behind their decisions [77]. Formally, given a model f, a data point x, and a prediction $f(x)$, it is becoming increasingly important to provide information about "why" or "how" the decision $f(x)$ was made. These recent constraints have led to the quick development of the field of eXplainable Artificial Intelligence (XAI), a subfield of AI interested in making machine learning models more interpretable while keeping the same level of generalisation performance [12]. The two main paradigms found in the literature are the training of inherently interpretable models, and the use of post-hoc explanation, *i.e.* add-on techniques that allow to interpret any complex model after it has already been trained.

Training inherently interpretable models is a herculean task because of the so-called accuracy-interpretability trade-off, *i.e.* the empirical observation that black box models tend to outperform interpretable ones. However, efforts are currently being made in the literature to develop novel interpretable models with state-of-the-art performance. For example, [105] proposed new SAT-based solutions to learn Decision Sets, a task that is known to be NP-hard. Moreover, a novel CNN architecture called ScenarioNet introduces the notion of scenarios: sparse representations of data encoding sets of co-occurring objects in images [46].

Although research on training inherently interpretable models is being conducted in XAI, our observation is that post-hoc explanations are used more often. The main appeal of these techniques is that, by their add-on nature, they can potentially increase transparency of state-of-the-art models, without resulting in a performance decrease, henceforth circumventing the accuracy-interpretability trade-off discussed previously. Post-hoc explanations can be placed in two main categories: model-agnostic and model-specific.

Model-Agnostic

These post-hoc explanation techniques provide information about the decision-making process of arbitrary model, *e.g.* black-box models. In this context, the explainer is only able to query the black box f at arbitrary input points z in order to provide an explanation. The most common model agnostic post-hoc explainers are locals surrogates, which are interpretable models trained to locally mimic the black box around the instance x at which one wishes to explain the decision $f(x)$. Formally, a sampling distribution N_x is chosen to represent a neighborhood around the instance x and the local surrogate g_x is taught to mimic f around x by minimizing the neighborhood infidelity

$$g_x = \min_g \mathbb{E}_{z \sim N_x} \left[(f(z) - g(z))^2 \right]. \tag{6}$$

The surrogate model g_x being interpretable, it can be used to provide a post-hoc explanation of $f(x)$. The most fundamental explainer of this type is called Local Interpretable Model-agnostic Explanations (LIME), and fits a sparse linear model g on f in a locality N_x that is specific to either tabular, textual, or image data [187]. The applicability of LIME to several types of data is one of the main reasons for its widespread use. However, alternative local surrogate explainers have recently been develop, and differ on their choices of interpretable models g and/or the neighborhood sampling distribution N_x. For example, LORE used decision trees as surrogate models g and a genetic algorithm for the sampling distribution [88, 171]. On the other hand, DeepVID locally fits a linear regression g using a Variational Auto-Encoder (VAE) to sample points z in the vicinity of x. [238]. Finally, LEMNA uses a mixture regression as g and samples from N_x by randomly nullifying components of the input vector x [90].

Alternative model agnostic post-hoc explainations are diagnostic curves such as Partial Dependency Plots and Adaptive Dependency Plots, which aim at computing two-dimensional representations of the global behavior of the black-box model along specific directions in input space [106].

Model-Specific

Such post-hoc explanations are restricted to a specific model type, *i.e.* tree-based models, neural networks, etc. In the context of image classification with CNNs, common explanations are saliency maps, which consist of images $x' \in \mathbb{R}^{3\times H\times W}$ that illustrate the pixels of the original image x that are the most important to the prediction $f(x)$. In [236], saliency maps were obtained by applying a mask $m \in [0,1]^{3\times H\times W}$ over the original image

$$x' = m \odot x + (1 - m) \odot r, \tag{7}$$

where r is a reference image (often chosen to be zero-valued) and $\odot$ is the component-wise product. These masks m can be obtained via a so-called *preservation game* where the smallest amount of pixel are retained while keeping a high class-probability, or a *deletion game* where the class-probability is considerably reduced by masking off the least amount of pixels. The optimization procedure used to compute the masks is very similar to adversarial attacks (see **Section 5.1**), therefore there is concern that the saliency map contain adversarial evidence *e.g.* artifacts of the optimization that do not highlight meaningful features from the image. To address this, the authors introduce a novel defense technique: by using clipping layers, they forbid neurons that were inactive on the original image x to become active on the masked image x', ensuring that only a subset of the high level features identified on the original image (edges, corners, textures, shapes etc.) are used when optimizing the masks. This defense has the added beneficial effect of providing fine-grained saliency maps. In [166], saliency maps were computed for charging post detectors using the specific Faster R-CNN network architecture. Their post-hoc method extracts information from a submodule of the Faster R-CNN network architecture called the Region Proposal Network. For tabular data, [4] combined a model-specific post-hoc explanation called *Layer-wise Relevance Propagation* in tandem with fuzzy set theory in order to provide textual explanations of a multilayer perceptron.

Other attempts to provide post-hoc explanations of neural networks work directly at the neuron level by looking at neural activation patterns in supervised learning [157] and reinforcement learning [158] as well as the high level features learned in deeper layers of a CNN [126].

Evaluation Metrics

One major difficulty in evaluating and comparing post-hoc explanations is the lack of ground-truth on what is the true explanation of a decision made by a black box. The amount by which an explanation is truly representative of the decision process of a model is called its "faithfulness". Note that faithfulness is a non-issue when using inherently interpretable models since explanations can be extracted directly by inspecting the model architecture (weights of a linear model, nodes of a decision tree etc.). Faithfulness is only pertinent to consider for post-hoc explanations of black box models, where we have no guarantee that the explanation has truly captured the "reasons" behind specific decisions. We now describe some of the attempts to define faithfulness of some post-hoc explanations.

When computing model-agnostic explanation extracted with local surrogate models, *i.e.* simple models that attempt to mimic the complex model near the point x of interest, faithfulness can be measured in terms of infidelity, see Equation 6. Large infidelities indicate that the local surrogates are unable to properly mimic the complex model and their explanations can safely be discarded. However, when comparing several local surrogate models with small infidelities, it is hard to state which one provides the most meaningful explanations. This is especially hard when the local surrogates are not the same types of models (linear for LIME, decision tree for LORE). Moreover, the infidelity metric depends on the choice of a neighboring distribution N_x, which is different for every method. For these reasons, one cannot use Equation 6 to compare post-hoc explanations from local surrogate models, although very large infidelities are valid ways to identify aberrant explanations.

For saliency maps of CNN classifiers, a measure of faithfulness consists of iteratively zeroing-out pixels in order of importance and reporting the corresponding decrease in class-probability output. The AUC of the resulting curve is used as a measure of how unfaithful to the model the saliency map is [236]. Zeroing-out a pixel value is meant to remove all information related to that specific pixel. This metric is promising and it would be interesting to adapt it to other types of data such as tabular and textual.

A promising way to compare/evaluate explanations is to measure their usefulness in debugging the model and in the ML pipeline in general. For example, in [166], the saliency maps of an electric charger detector were used to understand where the model was putting its attention when it failed to detect the right objects, which helped engineering specific data augmentations. Notably, the network was found to put a lot of its attention on pedestrians in the background, which was tackled by introducing negative examples with only pedestrians and no electric charger. This is a great practical example of post-hoc explanations being used to design the ML pipeline.

- Interpretability is quickly becoming an important part of machine learning systems.
- The two main paradigms of eXplainable Artificial Intelligence are the training of intrinsically interpretable yet accurate models, and the use of post-hoc explanations which aim at explaining any complex model after it is trained.
- There is currently no universal metric to assess whether or not a post-hoc explanation has truly captured the "reasons" behind specific decisions made by a black box. We are indeed very far from tackling the EU regulations on the "right to an explanation". Therefore, we think that a more realistic short-term goal is to evaluate explanations on their usefulness in designing the ML pipeline and in debugging complex models, similar to the experiments in [166].

5.4 Verification

We define Verification as any type of mechanism that allows to test formally or not that a given model respects a certain number of specifications. As such, it differs from method seen in the **Section 5.1** in the sense that the methods are not used to improve a model resilience, but rather to evaluate a certain number of property (which can be linked to robustness for instance) *after* the model was refined and/or trained. Hence, it doesn't act on the model but rather aims at verifying it. Most of the approaches are based on test input generative approaches, that is techniques that aim to generate failure inducing test samples. The idea is that, through careful selection of such samples, it's possible to cover "corner-cases" of a dataset/model which are examples for which the model wasn't trained on or doesn't generalize well enough to the point it would fail to correctly predict on such inputs. In a sense, such examples can be considered a part of the distributional shift problem. We distinguish two main approaches: one that is based on formal methods, that is a complete exploration of the model space given a certain number of properties to check, and one that is based on empirically guided testing, using criteria or relations that can lead to error inducing inputs.

5.4.1 Formal Method

Formal methods provide guarantees on a model given some specifications through mathematical verification. The increasing use of ML on safety-critical applications has led researchers to derive formal guarantees on the safety of those applications.

Model Checking is a technique used to formally verify a system given a set of desirable properties. The verification procedure exhaustively explores the state space of the model to determine whether the property is satisfied or not. Those verification techniques provide formal guarantees regarding a system's desirable properties. During the verification process, the desirable property is translated into first-order logic, temporal logic or a domain-specific (arithmetic, arrays, bit-vectors, datatypes) formula. This formula is satisfiable if and only if there exists a model of the system that satisfies it. In the literature, the verification of neural networks has been done through model checking or via frameworks that incorporate model checking.

Verification through Reluplex

Satisfiability Modulo Theory (SMT) based Model checking has been implemented on machine learning applications, mostly to verify neural networks. Notably, [118] proposed Reluplex, an SMT-based framework that verifies robustness of neural networks with RELU activations using a Simplex algorithm. In its most general form, Reluplex can verify logical statements of the form

$$\exists\, x \in \mathcal{X} \quad \text{such that} \quad f(x) \in \mathcal{Y}, \tag{8}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are convex polytopes, *e.g.* sets which can be expressed as conjuctions of linear inequalities. Reluplex can either assert that the formula holds (`SAT`) and yield the specific value of x that satisfies it, or it can confirm that no such point x exists (`UNSAT`). This framework outperforms the existing ones in the literature as it allows to verify neural network robustness in much less time. However, as the name suggests, the method is restricted to ReLU activations, and it can only verify robustness with respect to the L_1 and L_∞ norms (because their open balls are polytopes).

Another issue is that Reluplex cannot be used to verify global robustness *e.g.* robustness at all points x. To tackle this issue, [79] developed a custom KMeans clustering algorithm to regroup data points in dense regions with unique labels. Reluplex was then used on each individual cluster in search for adversarial examples which can be labeled differently from the rest of the cluster. They successfully identified robust (safe) clusters on ACAS XU and MNIST. The results are promising but more datasets should be experimented on to ensure that their KMeans algorithm is reliable and versatile enough.

Reluplex was also used in aircraft collision avoidance systems (ACAS) [114], where safety can only be assured if the advisory provided by a DNN can always lead to the avoidance of a collision with a second plane. To assert such a statement, the authors used Reluplex to verify the existence of instances, where the DNN gave an unsafeable advisory, when a safeable advisory existed. These counterexamples were then used to improve the safety of such neural networks. The downside of the proposed methodology is that the safety criteria are too hard for DNNs to achieve. As such, the authors plan to relax them by considering a region safe as long as it is safe by at least two advisors.

In [121], the authors find invariants in DNN based on decision patterns, which are specified as an activation status for a subset of neurons. The method is similar to the verification method based on Reluplex. For a given input, they try to find the region of space that is activated or not. On the one hand, [244] adopted Reluplex and a robust control theory in order to certify a neural network inside a closed system (dynamic settings). The neural network represents the policy of the dynamic system with unknown uncertainty. On the other hand, [115] combines Adaptive Stress Testing (AST) with Reluplex verification to determine the input perturbations that would push a simulated plane away from the track lane.

A framework similar to Reluplex is Reluval [240], a system for formally checking security properties of Relu-based DNNs. ReluVal can verify a security property that Reluplex deemed inconclusive. But to improve its robustness verification, [182] upgraded ReLuval by adding Quantifier Estimation to compute the range of activations of each neuron. Their method benefits from parallelization as neurons from one layer share the same inputs. Their experiments show they can verify robustness in small neighbourhoods. Using overapproximation and/or longer calculation time they can achieve this for bigger neighbourhoods. [113] present an approach for reachability analysis of DNN-based aircraft collision avoidance systems by employing Reluplex and Reluval verification tools to over-approximate DNNs. By bounding the network outputs, the reachability of near midair collisions (NMACs) is investigated. Looseness of dynamic bounds, sensor errors and pilot delay were investigated, in the experiments addressing all issues with real-time costs. However, the approach needs to be tested on real aircraft collision avoidance systems, to better assess its effectiveness.

SMT-based Model checking: Direct Verification

In [104], SMT-based verification process was applied on image classifiers subject to adversarial attacks. The process of verification relies on discretizing the region around the input to search for adversarial examples in a finite grid. It proceeds by analyzing layers one by one. The results of the verification are either the NN is safe with respect to a given manipulation or the neural network can be falsified. The latter is provided to a tester for fine-tuning the neural network. The implementation of the algorithm is done through the software Z3 which implements SMT verification. The approach gives promising results but suffers from its complexity, and the verification is exponential in the number of features. [164] propose FANNET, a formal analysis framework that uses model checking to evaluate the noise tolerance of a trained neural network. During the verification process, in case of non-satisfiability of the input property, a counterexample is generated. The latter can serve to improve training parameters and workaround the sensitivity of individual nodes. The experiments successfully show the effectiveness of the approach, which is however limited to fully connected feed-forward neural networks.

Bounded and statistical Model Checking

A sub-field of model checking is statistical model checking, SMC *e.g.* a technique used to provide statistical values on the satisfiability of a property. [18] quantifies how robust a neural network is through its performance against adversarial inputs. The proposed approach provides PAC-Style guarantees regarding the satisfiability of logical properties specified over the neural network. The authors focused their attention on binarized neural networks where both networks and their properties can be encoded using logical formulas. Both the neural network and the properties are implemented in their tool NPAQ which quantitatively does the verification. The approach has applications in security with respect to trojan attacks and to evaluate how fair is the prediction of a neural network. In [83], the authors study Deep Statistical Model Checking. We have an MDP which describes a certain task and a neural network trained on that task. The trained neural network is considered as a black-box oracle to resolve the non-determinism in the MDP whenever needed during the verification process. The approach is evaluated on an autonomous driving challenge where the objective is to reach the goal in a minimal number of steps without hitting a boundary wall. Another subfield of model checking that can help the implementation of machine learning on safety-critical applications is bounded model checking. The latter consists of constructing Boolean formulas that are not satisfiable if there is a counterexample of length k. [203] propose a verification framework based on incremental bounded model checking on neural networks. The framework uses CUDA. To detect adversarial cases on neural networks, two verification strategies are implemented: the first strategy is an SMT model checking with a model of the neural network and some safety properties of the system. The second strategy is the verification of covering methods. A covering method can be seen as an assertion that measures how adversarial two images are. During the verification process, the properties can be verified or a counterexample is produced.

Reachable sets

Given an input set I (possibly specified in implicit or abstract form, *e.g.* polyhedron, star set, etc.), the corresponding reachable set is defined as the set $\{f(X) : X \in I\}$ of all outputs of the neural network f on inputs from I. Having access to reachable sets of a DNN for suitable input sets provides important information on the behavior of the model which can be leveraged to verify its safety. However computing reachable sets of a DNN is a difficult task. There is no universal algorithm to compute them and over-approximations of reachable sets are sometimes considered for tractability or efficiency. Some methods involve computation over layers by applying sequence of activation functions, others involve deep analysis of the input set. [226] propose NNV (Neural Network Verification) a verification tool that computes reachable sets of DNNs to evaluate their safety. Under NNV, a DNN with RELU activation functions is considered to be safe if its reachable sets do not violate safety properties. Otherwise, a counterexample describing the set of all possible unsafe initial inputs and states can be generated. NNV benefits from parallel computing which makes it faster than Reluplex [118] and other existing DNN verification frameworks. Even though the approach performs well in terms of run time and efficiency, its effectiveness during the verification and the generation of counterexamples can be questioned in the case of learning-enabled CPS, such as closed-loop control systems. If the plant model is nonlinear, then at best one can overapproximate reachable sets, so any counterexamples generated could be spurious. In [225], the authors design three reachable computation schemes for trained neural nets, exact scheme, lazy-approximate scheme, and mixing scheme. On top of their techniques, they added parallel computing to reduce the run time of computing the exact schemes, which is performed by executing a sequence of StepRelu operations. The lazy-approximate is inexpensive and useful to quickly estimate the output range of feed-forward neural networks as it does not need to solve any linear optimization problems when constructing the reachable set. The mixing scheme combines the exact scheme and lazy-approximate scheme to give users a flexibility to obtain a tight enough output range for verification while maintaining the verification time under a specified threshold. The authors evaluated their approach on safety verification and local adversarial robustness of feedforward neural networks. [252] proposes a technique similar to these previous works, by Discretizing MLP input space into cells before applying a layer-by-layer computation to estimate the output set for all inputs with over-approximation. They show the effectiveness of the approach on a dynamic system. But propagation inside layers can yield to very conservative bounds and potentially huge computation costs with deeper networks.

Abstract Interpretation

In an effort to verify desirable properties of DNNs, researchers have studied abstract interpretation to approximate the reachable set of a network. With abstract interpretation, input sets and over-approximations of their reachable sets are specified using abstract domains, *i.e.* regions of space that are described by logical formulas. The verification process leverages the DNN approximation to provide guarantees. [71] studied Abstract Interpretation to evaluate the robustness of feed-forward and convolutional neural networks. The goal is to propagate abstraction through layers to obtain the abstract output. Afterwards, the properties to be verified are checked on that abstract output. This approach suffers from the quality of properties to be verified since they are based on few random samples. [135] propose to enhance Abstract Interpretation to prove a larger range of properties. The main contribution of the paper is to use symbolic propagation through neurons of the DNN, to provide more precise results in the range of properties that can be verified through abstract interpretation. In [197], abstract interpretations are used to verify the robustness of object recognition systems to input perturbations such as rotation and occlusion.

Linear Programming

Some verification techniques involve Linear Programming (LP). [141] proposes to verify the robustness of a DNN classifier by leveraging linear programming. The idea of the proposed technique is to find using nonlinear optimization, a suspicious point that is easier to be assigned a different label than other points and can mislead the classification process. The results of the optimization problem is a set of constraints which are solved via linear programming. The verification process will determine whether or not the DNN is robust against minimal perturbations of the input regions. In [191], a novel algorithm that computes more efficiently splits the neural network inputs to reduce the cost of verifying deep feed-forward RELU. For the splitting process to be effective, they estimate lower and upper bounds of any given node by solving linear programs. Also based on LP, [8] proposed an approach to certify ReLU-based networks by providing an upper bound of the relaxation error when using LP relaxation and the best two partitions that minimizes the upper bound. In other words, the paper shows first that partitioning is theoretically guaranteed to reduce relaxation error, potentially even giving zero error. As the number of parts in the exact partition can grow exponentially with the number of hidden units, the authors focus on using two-part partitions.They derive the two-part partition based on the form of the exact partition that minimizes the relaxation error upper bound. They have successfully tested their approach on a one-layer and a two-layer network trained on Iris dataset. [147], employed LP to provide certifiable robustness upper/lower bounds on neural networks. Given a classifier, the authors aim at obtaining a guarantee that the classification of an instance will not be changed if the instance is perturbed while being restrained to the L_p ball around the original instance. To do so, the authors compute a lower bound L on the output of the network corresponding to the predicted class, as well as an upper bound U on the output neurons of all other classes. If one has $U < L$, then the network is guaranteed to be robust to any small perturbations in L_p norm. LP was also used by [87] in order to automatically reduce the intent detection mismatch of a prosthetic hand.

Mixed-integer linear program

Verification of neural networks can also be modelled as a mixed-integer linear program (MILP). [181] propose to verify the satisfiability of the negation of specification rules on a trained neural network, both modelled as a MILP. Each specification rule indicates the desired output set for the input set. If a satisfying configuration is found, it will be considered as a counterexample violating the specification rules. The approach presented in that paper is an attempt to investigate the functionality certification of neural networks, according to defined specifications. MILP has been employed in several others studies in the literature. In [54], the authors employed MILP to estimate the output ranges of neural networks given constraints on the input. [37] used a MILP solver to find the maximum perturbation bound an Artificial Neural Networks can tolerate. This Maximum perturbation bound is defined as the norm of the largest perturbation which can be applied to an input that is strongly associated with a given class, while either maintaining the same predicted class, or keeping the probability of that class among the highest. For this approach, the perturbation bound does not apply when the input underwent an affine transformation. [206] relies on MILP to refine the lower/upper bound of activation for each layer, except they use LP relaxation and abstract interpretation on some layers. A certification property can be stated with the proposed approach: for all inputs in a specific region, the output activation corresponding to a specific class is always the largest. Experiments show comparable or even better results than compared state-of-the-art methods, especially on complete certification. Despite its successes in the verification of neural networks, MILP is restricted to piece-wise linear networks. Authors of [26] propose a general framework called Branch-and-Bound for linear and non-linear networks. This framework regroups several pre-existing verification techniques for neural networks. By defining a general framework, the authors are able to identify limitations and flaws of previous methods and implement their own improved method. They demonstrate the good performance of their branching strategy over various verification methods.

Gaussian processes verification

A Gaussian process is a stochastic process that can be used for regression and classification tasks. Their key feature is their ability to estimate their own uncertainty (aleatoric and epistemic), an information which can be leveraged for verification. In [208], the discussed approach provides a lower bound on the number of dimensions of the input that must be changed to transform a confident correct classification into a confident miss-classification. However, the experiments only considered Gaussian Processes with Exponential Quadratic kernels. Other kernels types should be investigated. Similar to this approach, [29] provides a theoretical upper bound on the probability of existence of adversarial examples for Gaussian processes. They also argue that because of the convergence of wide networks to Gaussian Processes, their bound has some applicability to DNN. This statement would need to be assessed with more experiments.

Formal verification: other methods

[223], explored the verification of decision trees and tree ensembles. They implemented a tool named VoTE (Verifier of Tree Ensembles), which has 2 components: VoTE core and VoTE Property Checker. The VoTE core will compute all equivalence classes in the prediction function related to the tree ensemble. VoTE Property Checker takes as input the equivalence classes and the property to be verified, and checks if the input-output mappings from each equivalence class are valid. The experiment conducted by the authors consists of evaluating the robustness, scalability and node selection strategy of VoTE. The latter

shows successful performance. Game theory has also been studied to evaluate the robustness of a DNN. [251] introduces a game theoretic approach relying on a two-player game to find an approximation to the maximum safe radius (w.r.t some dataset) and estimate feature robustness. Given an input, Player I selects by turning a feature to perturb and Player II chooses a perturbation. While Player II aims at minimizing the distance to adversarial examples, the game is cooperative for the maximum safe radius approximation and competitive for the feature robustness. Moreover, they use Lipschitz continuity to bound the maximum variation on outputs depending on inputs in order to provide guarantee bounds on all possible inputs. To address the intractable nature of the computation space, the authors use an approximation relying on Monte-Carlo Tree search for the upper bound estimation and the path finding A^* algorithm with pruning for the lower bound. Unfortunately, the lower/upper bound approximation gap widens as the number of features increases. In [190], the authors provide global robustness approximation sequences for lower/upper bounds using Hamming Distance, since the classical safe radius with L_0 norm is NP-hard. Their approach offers provable guarantees and an effective and scalable way of computing them. However, the experiment section of the paper would appear to lack coherence, which makes the approach hard to evaluate. A hybrid approach to verifying a DNN has been studied in [241]. The authors propose to estimate the sensitivity of neural networks to measure their robustness against adversarial attacks. The sensitivity metric is computed as the volume of a box over-approximation of the output reachable set of the neural network given an input set. They applied two methods to compute the approximations, a dual objective function representing the sensitivity computed using the dual formulation and the sensitivity computed via Reluplex [118]. However, it is not clear what kind of adversarial attacks the robustness criteria might prevent from. [72] addressed the verification of a DNN with a Bayesian approach. They provided a framework that uses Bayesian Optimization (BO) for actively testing and verifying closed-loop black-box systems in simulation. More specifically, BO is used to predict the environment scenarios and the counterexamples that are most likely to cause failures in the designed controllers. They have tested their approach on some simple functions and then on some benchmark environments of OpenAI Gym. Regarding the application domain, it is unclear what kind of specification could be covered by the approach.

Although failure of neural networks predictions can occur in classification, there exists nuance between the different types of errors that can be made by the model. This nuance is investigated in [196], who propose a safety analysis framework called Classification Failure Mode Effects Analysis (CFMEA), which identifies the relevant failure modes using an abstraction of the perception-control linkage of the autonomous driving system. Relying on a *classification hierarchy* , they identify four possible outcomes in a classification task, based on whether the input is correctly classified to the “best” label: correct classification (*e.g.* correctly predict a “car”), under-classification (*e.g.* correctly predict a “vehicle”), misclassification (*e.g.* incorrectly predict a "truck"), and under-misclassification (*e.g.* incorrectly predict "other"). The authors introduced equations to calculate the risk and control policy, as well as the action severity based on these two metrics. The result and the analysis process could be very useful for autonomous car producers to improve their machine learning systems. During experiments, the authors also observed that even a simple classification architecture can lead to a large number of classification cases because of the under-classification. Thus a complexity reduction strategy needs to be introduced in the future. However, This work can be improved in some aspects:

- The choice of non-leaf classes can lead to different results in terms of safety analysis. Especially, the complexity of the classification cases and the progress (time to complete a task) can vary much based on if we merge two leaf classes or not.
- The proposed safety analysis framework is very interesting but still needs to be verified in a more realistic scenario to validate how it can be generalized and whether/how it can handle a real-time data stream.

To assess the safety of aircraft systems, the authors of [63] study the verification of Boeing's neural network-based autonomous aircraft taxiing system. They defined a safety requirement as: in 10 seconds, the plane must reach within 1.5 m of the centerline and then stay there for the remainder of the operations. Failing to satisfy this requirement will be considered as a counterexample. In their experiment with the X-Plane flight simulator, they found only 55.2 % of the runs satisfying the requirement while 9.1 % of the runs completely left away from the centerline. They analyzed the failed cases and observed that cloud and shadow can misguide the plane's taxiing. In addition, the NN of the TaxiNet system poorly handled intersections. Using this diagnosis information, the authors retrained the NN and obtained 86% of successful runs and only observed 0.5% of runs leaving the runway. This approach is very promising to be used by other avionic companies to verify their autonomous AI or ML-based taxiing systems. It can also be used to verify other autonomous systems, such as landing, collision avoidance, or deicing systems. As the author mentioned, there are still 14% of the runs that failed to satisfy the safety requirement. To certify this taxiing system for real aircraft, we need to further improve the performance of the neural network. Another line of research aims to check the satisfiability of safety properties of neural networks by estimating tight output bounds given an input range [239]. This approach relies on symbolic linear relaxation to provide those tighter bounds on the network output. Then a directed constraint refinement process follows to minimize errors due to the relaxation process. The experiment's results show that the approach outperforms state-of-the-art analysis systems and can help improve the explainability of neural networks.

5.4.2 Testing Method

Testing activities are still a key part of traditional software systems. The most basic aspect of it is to input a value to a model for which we know the output that has to be returned (*an oracle*). Many more advanced methods can be used to test a model whether it is based on criteria such as code coverage related metrics, on specifications such as functional testing, on an empirical comparison such as differential testing or more, with possible combinations of different techniques. And in that regard, safety critical systems are no exceptions. Standards such as DO-178C even introduced testing methods as an integral part of the process with the MC/DC criteria, which aim to test combinations of variables yielding different output with only one factor changing. Hence, testing will likely be an integrated part for ML certification. However, because of the paradigm shift that ML systems introduced, previously used methods need to be adapted and revised to take into considerations such differences and new emerging methods will have to be developed to tackle the arising challenges.

Testing criteria

In the direct lineage of code coverage related criteria, a new set of ML related ones have been developed. [173] was the first to introduce the notion of *neuron coverage* (NC), directly inspired by traditional code coverage. Formally, given a set of neurons N, the neuron coverage of a test set T of inputs was originally defined as

$$\mathrm{NC}(T) := \frac{\#\{n \in N \,|\, \mathrm{out}(n,x) \geq \tau \ \ \forall x \in T\}}{\#N}, \tag{9}$$

where the symbol # refers to the number of elements in a set (cardinality) and $\mathrm{out}(n,x)$ is the activation of the neuron n when the input x is fed to the network. Simply put, the NC designates the ratio of "fired" neurons (*i.e.*, whose output is positive/past a given threshold) on a whole set of inputs. The main criticism associated with this measure is that it is fairly easy to reach a high coverage without actually showing good resilience, since it does not take into account relations between neurons as a pattern and discards fine grained considerations such as the level of activation by simply considering a boolean output. Following this first work, related criteria were developed to extend the definition and tackle those issues: KMNC (K-Multisection NC) [148] aimed to take into account "bands" of neurons activation to give a more detailed analysis of the coverage, Top-k NC [253] only considered the k neurons with the strongest activation within a certain layer, as they are more likely to have an impact down the line. Some other criteria try to take inspiration from other forms of traditional testing criteria while still focusing on neuron activation. T-way combinations [149], inspired from combinatorial testing, aims to quantify coverage for given t-way neurons configuration inside a layer, with [202] using a similar idea but with the goal to evaluate the configurations across two consecutive layers through the use of triplets of activation (how two neurons of a layer influence results in another layer), while Sign-Sign (and related) coverage [216] explored neurons activation based criteria that mimics MC/DC criterion by taking into account neurons activation state change in between layers and pairs of inputs.

Test generation coverage based

In general, criteria are not used as a plain testing metric like with traditional software, but rather as a way to incrementally generate test cases that maximize/minimize those given criteria. As most DNNs are trained on image datasets, it's fairly simple and straightforward to generate new images supporting an increasing coverage. To achieve this, techniques such as fuzzing process to randomly mutate base images sampled from a dataset [253][237][89][50] or greedy search [221] and evolutionary algorithm [21] coupled with transformation properties can be used. Those properties are common geometric and/or pixel based transformations such as rotation or contrast variation that are considered *invariant*, that is images affected by such a transformation shouldn't see their prediction altered through the network if the variations are small enough. This principle is linked to Metamorphic Testing, which aims to test two similar images, one having been modified with such invariant property. More complex methods can be used such as *concolic* testing [215] which mixed symbolic execution (linear programming or Lipschitz based) with concrete input to generate new test cases helped by heuristic based on coverage with improved results compared to other methods. Some techniques leverage Generative Adversarial Network (GAN) [265] or assimilated in order to generate more realistic test images through coverage optimization. Of course, if the image obtained tends to be more "natural" compared to fuzzy or metamorphic ones, these methods suffer from the traditional downsides specific to GANs and need extra data to work. Fairly few of those methods have been applied to transportation related datasets, DeepTest [221] being the most prominent example of application. Yet, the interesting results achieved on other datasets show the possibility to develop it in a multitude of directions thanks to different criteria and to apply it in transportation systems. Limiting factors of those techniques, aside from the choice of criteria, remain the time needed to obtain the generated test and the effective quality of the images. Some transformed images, while interesting for analysis purposes, might not yield effective insight as they would not happen in real life situations. Note that most, if not all, of the techniques use primarily the Neuron Coverage criteria for its straightforwardness of implementation, and because it is the closest to the idea of code coverage, but as we have mentioned previously, it has many limitations.

Test generation other methods

If coverage based generation is the most widespread technique used, we identified techniques utilizing other methods to manage test cases generation for testing purposes. On image datasets, techniques based on Adaptive Random test [256] leveraging PCA decomposition of network's features, Low-discrepancy among sequences of images and Active Learning [52], with metamorphic testing either using entropy based technique with softmax predictions [229] or through the search of critical images following those transformations [174]. Note that metamorphic testing [33] is an effective proxy for image generation and testing which is also used by techniques mentioned previously such as DeepTest. [249] used SIFT based algorithm to identify salient parts of the image and optimize for adversarial examples using a two-player game. [255] proposed a method to generate adversarial examples in a non-linear control system with a Neural Network in the loop, by deriving a function from the interaction between the control and the NN.

As for RL applications to transport related tasks, simulation environments are used for training models. In this context, testing methods consist of generating scenarios representing behavior of agents in the environment that are more likely to lead to failure of the learned policy. Meta-heuristic is the preferred method to generate procedural scenarios whether it be evolutionary [67] or simulated annealing with covering arrays [227]. Scenario configuration can also be tackled from the point of view of a grammar based descriptive system [250] which allows for a flexible comparison between cases.

Fault localization/injection testing

Fault localization is a range of techniques that aim to precisely identify what leads to a failure. In the same vein as traditional fault localization, some papers investigated the root of prediction errors directly on neurons in order to find out which neurons are involved in it, opening the door for potential new test cases generation or resilience mechanisms. Note that fault localization in this context is at the crossroad between *Verification* and *Explainability*, illustrating that methods are not restricted to one domain. Deepfault [56] was developed in order to identify exactly which pattern of neurons are more present in error inducing inputs, thus allowing a generation of failure inducing tests through the use of gradient of neuron activation on correctly classified examples.

Fault injection denotes techniques that voluntarily (but in a controlled way) inject faults in order to assess how the system behaves in unforeseen situations. In that setting, TensorFI [34] is a specialized tool that aims to inject faults directly in the flow graph of Tensorflow based applications. Note they provide support both for software and hardware (bits) level of faults.

Differential Testing

Classical testing methods for DNN classifiers require testing prediction of an input against a ground truth value. If this is possible for a labelled dataset, it's much more complicated when there are no known labels which arise especially during the operation phase, where the ML component is deployed in a live application. Here, we trust the performance of the network on the data distribution to predict in a relevant way. The existence of Adversarial (See **Section 5.1** *Robustness*) or Out-of-Distribution (See **Section 5.2.2** *OOD detection*) examples demonstrate that it is not the best strategy. In particular, the absence of ground truth, or "oracle", is known in software engineering as the *oracle problem.*

One way to circumvent the oracle problem is to introduce "pseudo" oracle to test correctness of an input, which is the main idea behind differential testing. By inducing a change on the model in a semantically equivalent way compared to the original, one can then test this equivalence on the prediction. A simple implementation of this process takes the shape of N*-versioning*, which consists in N semantically equivalent models that will be used to test an input. Since models are equivalent, so should their outputs if they receive the same input. In the case of DNN, the equivalence between models can be easily achieved by training them independently on the same dataset, possibly using a different framework, and to compare the predictions of the N models for a given input. N-versioning is strongly related to the notion of *ensemble* learning, which uses the knowledge of multiple models. In particular, [81] showed the advantage of ensemble learning against adversarial examples. In the case of pure N-versioning, NV-DNN [254] used this mechanism with a majority vote system in order to reject potential error, with the introduction of criteria to quantify percentage of errors that cannot be caught by the model (*i.e.* all models do the same error). The combination of models possible for N-versioning was also studied in more detail in [150] where the diversity of different architecture can bring over input error rejection/acceptance is explored. Indeed models do not need to differ only in their parameters, they can rely on various mechanisms as long as similarity is preserved. For instance, D2Nn [138] uses neurons more likely to contribute to errors in order to build a secondary network. The primary and secondary network can then be used for comparison of predictions within a threshold.

Yet, differential testing is not limited to semantically similar models, any semantic comparison allowing to build the “pseudo” oracle proxy is good. Hence, it’s possible to use specificity of DNN through neurons activations [35] to derive a “pseudo” oracle, by gathering activation patterns of the train data, and comparing them to new inputs pattern to reject or not the inputs if the patterns differ too much. Another technique investigated in [179] used pairs of spatially/temporally similar images for the comparison. Note that Metamorphic Testing mentioned in Test generation methods can be viewed as a form of differential testing. For those methods, the main limitation is determining the policy and mechanism in order to balance correctly between false negative and positive examples as well the semantic modifications. In particular, the change needs to induce diversity in the process while not being too different to preserve the similarity.

Quality testing

While traditional testing methods aim at testing the model (and so the data specifications indirectly), another interesting idea would be to test the data *directly*. Indeed, since the model is using this data to learn, if one could get a quality measure of the data used, it could add an extra layer to the certification process. We identify some techniques that focus on this approach: [154] introduced quality criteria for Convolutional Network based classifiers. The idea is to make use of the way data are distributed by studying the repartition across classes and distances from samples to the centroids/boundary of each class in order to evaluate deficiencies in the dataset. In particular, they demonstrate that traditional dataset can perform poorly based on such measurement, while yielding good accuracy. Similarly, but for bias/confusion testing, [222] defined metrics to quantify bias or confusion for classes of a dataset; the idea here is to test if the model learned the data fairly. In [5], a Probability of Detection (POD) method on binary classifiers is proposed in order to quantify how well the test procedure can detect vital defects. In a different direction, some techniques deal with test prioritization. Indeed, with the ever growing size of the systems, testing them can become quite expensive. As such, being able to select test samples, which are most likely to induce errors, is important. DeepGini [60] took the approach to quantify the relevance of test inputs, for prioritization purposes, through the likelihood of miss-classification of the model. The more classes likeliness are close, the less the model is confident, thus a 0.5/0.5 prediction for a two-classes classifier means the model has no confidence in its verdict. However, as we have seen in Section 5.2, one needs to be careful when using softmax output for confidence comparison for data that lies far from the training distribution. In [3] test instances were instead clustered based on their semantic and test history and some statistical measure based on similar test failures to prioritize most important test samples is used. While in [74], datasets were analyzed to extract relevant features and combinatorial testing is used in order to cover as many cases as possible with a small number of tests. Those approaches share the interesting perspective of trying to test directly relations between the model and the data, especially with respect to their quality, where traditional metrics rely on accuracy. However, those methods rely mostly on a hand crafted metric together with some empirical measures and/or heuristics. While information obtained is interesting, the metric itself might not be generalizable depending on the model/dataset.

- Formal verification for ML can suffer from combinatorial explosion regarding the size and the complexity of the model to be verified.
- The verification process requires a formal representation of the system, which is subject to interpretation as to which best describes the system.
- Testing methods for ML certification generally reuse traditional methods such as coverage based testing or differential testing, while taking into account ML specificity to adapt them.
- Most of those methods however are based on empirical considerations or ad-hoc observations, hence they would benefit from either a more theoretical approach to derive criteria or a combination with more formal methods presented in the previous section to bolster their effectiveness.

5.5 Safe Reinforcement Learning

Suppose that an agent interacts with an environment by perceiving the state of the environment, performing actions and then receiving a reward signal from

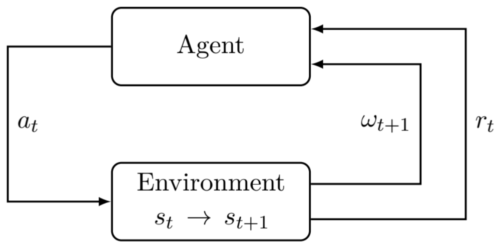

Fig. 10 Agent interacting with its environment [62].

the environment. The main task here consists of learning how to perform sequences of actions in the environment to maximize the long-term return which is based on the real-valued reward. Formally, the RL problem is formulated as a discrete-time stochastic control process in the following way: At each time step t, the agent has to select and perform an action $a_t \in A$. Upon taking the action, (1) the agent is rewarded by $r_t \in R$, (2) the state of environment is changed to $s_{t+1} \in S$, and (3) the agent perceives the next observation of $\omega_{t+1} \in \Omega$. Fig. 10 illustrates such agent-environment interaction. An RL agent aims at finding a policy $\pi \in \Pi$ that maximizes the expected cumulative reward or *return*:

$$\begin{aligned} &V^{\pi}(s) = \mathbb{E}[\, R_t | s_t = s\,], \\ &\text{with } R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \end{aligned} \tag{10}$$

where $\gamma \in [0, 1]$ is a discount factor that applies to the future rewards. A deterministic policy function indicates the agent's action given a state, $\pi(s) : S \rightarrow A$. In the case of stochastic policies, $\pi(s, a)$ indicates the probability of choosing action a in state s by the agent.

Recently, researchers have successfully integrated DL methods in RL to solve some challenging sequential decision-making problems [75]. This combination of RL and DL is known as deep RL, and benefits from the advantages of DL in learning multiple levels of representations of the data to address large state-action spaces with low prior knowledge. For example, a deep RL agent has successfully learned from raw visual perceptual inputs including thousands of pixels [161]. Deep RL algorithms, unlike traditional RL, are capable of dealing with very large input spaces, and indicating actions that optimize the reward (e.g., maximizing the game score). As a consequence, imitating some human-level problem solving capabilities becomes possible [68, 162].

Value-based approaches

The value-based algorithms in RL aim at learning a value function, which subsequently makes it possible to define a policy. The value function for a state is defined as the total amount of discounted reward that an agent expects to accumulate over the future, starting from that state. The Q-learning algorithm

[245] is the simplest and most popular value-based algorithm. In its basic version, $Q(s,a)$ with one entry for every state-action pair is used to approximate the value function. To learn the optimal Q-value function, the Q-learning algorithm uses a recursive approach which consists in updating Q-values based on Bellman equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[\, r_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1}, a') - Q(s_t, a_t)\,], \tag{11}$$

where α is a scalar step size called the learning rate. Given Q-values, the optimal policy is obtained via:

$$\pi(s) := \arg\max_{a \in A} Q(s, a).$$

The idea of *value-based deep RL* is to approximate the value function by DNNs. This is the main principle behind deep Q-networks (DQN), which were shown to obtain human-level performance on ATARI games [161].

Policy gradient approaches

Policy gradient methods maximize a performance objective (typically the expected cumulative reward $V^\pi(s)$) by discovering a good policy. Basically, the policy function is directly approximated by a DNN meaning that the network output would be (probability of) actions instead of action values. It is acknowledged that policy-based approaches converge and train much faster specially for problems with high-dimensional or continuous action spaces [2]. The direct representation of a policy to extend DQN algorithms for addressing continuous actions was introduced by Deep Deterministic Policy Gradient (DDPG) [140]. This algorithm updates the policy in the direction of the gradient of Q which is a computationally efficient idea. Another approach is using an actor-critic architecture which benefits from two neural network function approximators: an actor and a critic. The actor denotes the policy and the critic is estimating a value function, e.g. the Q-value function.

Safety in RL

In safety-critical environments, researchers concentrate not only on the long-term reward maximization, but also on damage avoidance since the agent does not always perform safe actions that may lead to hazards, risks, and accidents in the environment. For example, in autonomous driving, an unsafe action may lead to a collision between the ego vehicle and other vehicles, pedestrians, or another object in the environment. From a theoretical perspective, a high-level argument on safety of RL-based systems was reported in [25] and could be used as the basis for the safety case of RL. The authors have identified and analyzed the technical and socio-technical factors that can affect the potential strength of the reasoning in ML. They stated that the traditional approach to safety which assumes a deterministic (predictable) system would not work for ML in general and RL in particular. They have suggested an “adaptive” approach to safety and discussed different aspects from a theoretical perspective. In another theoretical tutorial

[210], the limit-reachability technique was employed to translate logic-based requirement specifications, omega-regular [91], into scalar reward that can be used by any model-free RLs. Consequently, this approach enables model-free RLs to produce a policy that maximizes the probability of satisfaction of the specification and then behaves safely. “Limit reachability” provides an alternative model-checking algorithm for omega-regular requirements of Markov Decision Processes and a foundation to verify more expressive specification. Experience-Based Heuristic Search (EBHS) [23] extended the idea of combining RL with search-based planning to continuous state spaces by improving heuristic search-based path planner using learned experiences of deep RL. Actually, the pretrained Q-values (state-action values which estimate reward for each action in each states of the environment) are integrated into a hybrid $A^\star$ (a traditional heuristic search algorithm) planner to replace the commonly used heuristic functions. Particularly, authors have applied Double Deep Q-Networks and “learning from demonstration” which is helpful when dealing with sparse rewards and high-dimensional state spaces. They have applied Deep Q-learning from Demonstrations (DQfD) for learning from demonstration which is based on pretraining from an expert policy. They have shown computational advantages and reliability of such an approach for two application types, in the field of path planning. The first one is about learning a non-holonomic heuristic (considering the kinematic constraints not obstacles which is the case for holonomic heuristic) estimating the optimal path onto a quadratic Bezier curve. The second is about a unified heuristic for a standard parking scenario. However, the evaluated scenarios are too simple compared to real world settings.
In this paper, we identify three main categories concerning approaches to safe RL:

- **Post-optimization**: Adding an additional safety layer after the RL to exclude unsafe actions like safe lane merging in autonomous driving [92],
- **Uncertainty estimation**: Estimating what the agent does not know in order to avoid performing certain actions, making the agent behaviour robust to unseen observations. For example, collision avoidance for pedestrian [146],
- **Stability certification of RL-based controllers**: Providing theoretical guarantees for RL-based controllers like [165].

5.5.1 Post-optimization

The idea behind these approaches is to add an additional safety layer after RL to exclude unsafe actions. WiseML was proposed in [152] to combine ML with runtime monitoring to detect violations of system invariants in the ML-based actions execution policies, *i.e.* RL, using a monitoring module. WiseML receives as input the environment variables and a set of goals to be achieved. It performs actions in the environment to reach the desired goals. The monitor module is in charge of blocking actions that will violate the system invariants, and give feedback to the RL module so that it can adapt itself. However, WiseML has remained a theoretical approach since no experiments are reported in the paper.

A preliminary approach was presented in [120] where authors have studied the safety of a modified version of DQN that is equipped with dropout, Q-network ensembling and a classifier to identify dangerous actions in simple gridworld environments. The modified version is reported to have less catastrophic outcomes. The same approach was tested in the CoinRun environment [40], a procedurally generated game which is able to measure an agent's ability to transfer its experience to novel situations. Overall, they have found that the uncertainty information conveyed by an ensemble of agents, *i.e.* in the value function, is helpful for predicting a catastrophe. In another work, authors have employed a policy-based RL algorithm, i.e. soft actor-critic, to solve complex scenarios in autonomous car driving, such as merging lanes with multiple other vehicles present [92]. To make it applicable in safety-critical applications, they have used a non-linear post-optimization to optimize the RL policy. The post-optimization computes a single homotopy class and the final output is an optimal control input sequence that generates the optimal state-space trajectory. After the post-optimization, an additional collision-check is performed to check for a high level of safety. Experiments were performed using Divine-RL, an autonomous driving simulation environment and results showed that the collision rate dropped to almost zero for the given time-horizon. However, authors did not compare their approach to any alternative approaches, so the effectiveness could not be assessed fairly. Moreover, the method should be evaluated using real road traffic not only by simulation. A dynamically-learned safety module has been proposed for RL agents in [16]. The module is a recurrent neural network trained to predict whether future states in the environment can lead to an accident or a collision. Experiments were conducted in a simulation environment with an autonomous driving car on a highway that can be surrounded by other traffic vehicles. Results showed that the proposed approach significantly reduces the number of collisions. However, this approach combines both hand-crafted rules and the learned behavior (from the history of the system). Since hand-crafted rules cannot behave properly in unexpected situations with a lot of changes, the applicability of the approach in such situations is questionable. In [107], authors have used a prediction mechanism for safe intersection handling in a RL-based autonomous vehicle. The prediction mechanism aims both at minimizing disruption to traffic (as measured by traffic braking), while avoiding collisions, and at the same time, at maximizing distance to other vehicles, while still getting through the intersection in a fixed time window. These optimizations generate different behaviors for the ego-car, but RL finally learned them with zero collisions. Unfortunately, the authors did not report any comparison of their approach to others. Although they presented computational complexity and ran the system on a real autonomous vehicle, no running time result was presented. According to our personal communication with authors, the proposed approach has been upgraded afterward to work at the speed of 10Hz that meets real-time requirements.

Distinguishing task failures has been proposed in [167]. Two failures were identified: 1) misclassifications that do not violate the safety constraints, and 2) harmful failures. The approach consists in training a NN in an RL environment: adjusting the velocity of a robot to avoid moving obstacles. They tested their approach to reveal failures and extract information about when they occurred. Authors then proceeded to predict the outcome in new operating conditions and successfully manage to replicate them through experiments. Finally, they added a safety function to prevent harmful failures and show through experiments that it did almost nullify harmful failures. However, the probability distribution of testing and operating conditions was assumed to be known. Yet this assumption seems restrictive for real world problems (they can be calculated, for instance, by studying weather patterns in the environment). Moreover, they consider fully-observable environments, which is very rare in real world situations. The authors asserted that addressing partial-observability is a future challenging work.

5.5.2 Uncertainty estimation

Similar to estimating the uncertainty (see Section 5.2), some researchers attempted to estimate the possibility of agent's failure to prevent hazardous actions leading to such failure. For example, reliable evaluation of the risk of failure in RL agents has been studied in [230]. Authors highlighted limitations and computational cost of random testing to obtain adversarial inputs/settings mentioning that reliable risk estimation can exceed training costs. They have adopted an adversarial evaluation approach to focus on adversarially chosen situations but the obtained failures were rare. So, they proposed a continuation approach for learning a failure probability predictor to estimate the probability of agent failures given some initial conditions. The main idea is employing data gathered from some less robust agents which fail often to improve the learning. The proposed approach has been evaluated on two RL domains: 1) Driving domain: the agent (on-policy actor-critic) drives a car in the TORCS simulator and rewarded for driving forward without crashing, and 2) Humanoid domain: the agent (off-policy distributed distributional deterministic policy gradient-D4PG) runs a 21-DoF humanoid body in the MuJoCo simulator and rewarded for standing without falling. Two different settings have been tested: 1) failure search: efficient finding of inputs (initial conditions) that cause failures, and 2) risk estimation: efficiently estimating the failure probability of an agent. The approach is reported to be effective in both settings for two tested environments. Estimating uncertainty, *i.e.* estimating what the agent does not know, is found helpful to avoid unsafe actions in the literature. These approaches attempt to make the agent behaviour robust and then safe to unseen observations. [146] have employed model uncertainty estimation to develop a safe RL framework for collision avoidance of pedestrians. The main component is an ensemble of LSTM networks that was trained with dropout and bootstrapping to estimate collision probabilities. These estimations have been used as predictive uncertainty, to cautiously avoid dynamic obstacles. The results showed that the model knows what it does not know; the predictive controller employs the increased regional uncertainty in the direction of novel obstacle observations, to act more cautiously in some novel scenarios. This uncertainty-aware framework showed more robustness to novelties and behaved safer than an uncertainty-unaware baseline. However, the approach should be tested in real world navigation systems. Cautious Adaptation in RL (CARL) [262] is a general safety-critical adaptation task setting to transfer knowledge (or skills) learned from a set of non-safety-critical source environments (e.g., simulation environment in which failures do not have heavy cost) to a safety-critical target environment. First a probabilistic model is trained using model-based RL to capture uncertainty dynamics and catastrophic states of source environments. Afterwards, in a new environment with unknown dynamics, the CARL agent attempts to avoid risky actions leading to catastrophic states. This is achieved by modifying the "action score" of PETS (a model-based RL approach using an ensemble of probabilistic dynamics models). The action score determines the mean of predicted rewards of an action sequence. Although the results are promising (one of tested environments is Duckietown car driving), there are some issues worth mentioning: 1) while CARL is reported to outperform other approaches, the authors did not report the running time of CARL: it needs a pretraining phase and its overall computational burden is not clear, 2) CARL is only applied to model-based RLs, and 3) The authors propose two CARL methods, but the effect of an important parameter is not investigated for one of the two methods proposed: the caution parameter. While it is set to 50 heuristically, its impact on the overall performance of CARL is not investigated, despite it being ablated for the other CARL method. Worst Cases Policy Gradients (WCPG) [220], a novel actor-critic framework, has been proposed to model uncertainty of the future in environments. In fact, WCPG estimates distribution of the future reward for any state and action and then learns a policy based on the uncertainty model (optimized for different levels of conditional Value-at-Risk). Therefore, the obtained policy is sensitive to risks and avoids catastrophic actions. To evaluate WCPG, they have considered two scenarios of unprotected left turn and merge into highway in a driving simulator. While WCPG can be adjusted dynamically to select risk-sensitive actions, it performed better compared to other state-of-the-art RLs in terms of collision rate. However, real-time execution of WCPG should be tested since the current experiments were performed using data that have been extracted from a driving simulator. WCPG must be deployed in the simulator to show its real-time effectiveness.

5.5.3 Stability certification of RL-based controllers

In [165], authors have provided quantitative stability conditions for a special kind of DL-based controllers in a closed-loop manner. The controller is a non-autonomous input-output stable deep neural network (NAISNet), which consists of a residual neural network composed of several blocks where the input and a bias is fed to each layer within a single block. This network has been used to approximate a baseline controller, which can be a human controlling the system. The authors have presented an explicit equation to compute a lower bound on the number of hidden layers that leads to a stable learning-based closed-loop system. They have employed Lyapunov functions, a classic approach for analyzing stability of dynamical systems in control theory. Although they mention autonomous driving applications as an example of human controller systems, their experiment was conducted on a very simple controller, namely a continuously stirred tank reactor. In any case, the experiment part is very limited and the approach seems to be non-practical. A probabilistic model predictive safety certification (PMPSC) scheme was proposed for learning-based controllers in [235]. This approach is designed to equip any controller with some probabilistic constraint satisfaction guarantees. They have combined Model Predictive Control (MPC) with RL to achieve safe and high performance closed-loop system operation. Their definition of a safe action (or output of the controller) is different; a learning-based action is certified as safe, if it leads to a safe state, *i.e.*, a state for which a potentially low-performance but online computable and safe backup controller exists, for all future times. They have considered possibly nonlinear stochastic systems that could be formulated as linear systems with bounded model uncertainties, and possibly unbounded additive noise. They have successfully tested their approach to learn how to safely drive a simulated autonomous car along a desired trajectory, without leaving a narrow road. For car simulation, they have used an a priori unknown nonlinear, time-invariant discrete-time dynamical systems described by equations of state-space representation. The approach appears more theoretical than practical, and so were the experiments which all consisted of simulations. To provide safety in terms of stability in a continuous action space environment, Lyapunov functions are leveraged to provide safety in [22]. The idea behind the approach is to extend Lyapunov stability verification to statistical models of the dynamics, to obtain high-performance control policies with provable stability certificates. Moreover, they showed that one can effectively and safely explore the environment in order to learn about the dynamics of the system and then employ such information to improve control performance and expand the safe region of the state space. They have successfully applied their algorithm to a simulated inverted pendulum stabilization task with safety guarantees. A modified version of the classical policy iteration algorithm has been presented in [32], to preserve safety by constraint satisfaction. The key idea is to compute control policies and associated constraint admissible invariant sets, that both ensure the system states as well as control that inputs never violate design constraints. Approximate dynamic programming (ADP) was employed to update control policies and associated constraint invariant sets iteratively, while ensuring the states of the system and control inputs never violate constraints. The preliminary results revealed that asymptotic convergence of the sequence of policies to the optimal constraint-satisfying policy is guaranteed.

In another work, authors have proposed to use projection to ensure that an RL process is safe without disrupting the learning process [82]. It is based on direct minimization of the learning equation under some safety constraints. Model Predictive Control (MPC) techniques are used by the authors to compute the safe set of states. The technique was applied on Q-learning and policy gradient for a simple system. However, the paper lacks some serious experiments to evaluate their proposed technique. The idea of Bayesian MPC has been introduced in [233]. Then authors have extended this idea to a theoretical framework for learning-based MPC controllers and have proposed a modified version that introduces cautiousness [234]. This cautious Bayesian MPC formulation uses a simple state constraint tightening that relates the expected number of unsafe learning episodes, which could be defined particularly for each system or scenario, to the cumulative performance regret bound. They have tested their approach on a generic drone search task, where goal is defined as collecting information about an a priori unknown position using a quadrotor drone. The safety-critical constraints combine a maximum range of the drone together with a minimum altitude that need to be satisfied under physical actuator limitations. The experiments performed are all based on simulation. The tested systems are defined as linear and nonlinear systems.

The idea of verification-preserving model updates has been introduced in [65]. The paper aims at obtaining formal proofs for RL in settings where multiple environmental models must be considered. For this purpose, authors presented an approach using a mix of design-time model updates and runtime model falsification for updating an existing model while preserving the safety constraint. They have successfully achieved formal safety proofs for autonomous systems (like Adaptive Cruise Control in cars) performing in heterogeneous environments.

Some approaches guarantee safety of RL in specific types of problems. For example, constrained cross-entropy [247] addressed safe RL problem with constraints that are defined as the expected cost over finite-length trajectories. The constrained cross-entropy generalizes the cross entropy method for unconstrained optimization by maximizing an objective function, while satisfying safety requirements on systems with continuous states and actions. For each iteration, it uses the set of constraints or safe requirements to sample a set of elite policies to update the policy distribution.

Logically-Constrained Reinforcement Learning (LCRL) [93] is a general framework that guarantees the satisfaction of given requirements and guides the learning process within safe configurations in high-performance model-free RL-based controllers. Authors have used Linear Temporal Logic (LTL) to specify complex tasks encompassing safety requirements. Then a reward function was automatically shaped to make sure that RL generally did not violate the property during and after learning. Moreover, they have employed an adaptive safe padding mechanism to balance the trade-off between efficient exploration and performing safely during learning. LCRL has been successfully evaluated on a set of numerical examples and benchmarks, including NASA

Opportunity Mars-rover, LCRL has certified learning outcomes in terms of the probability of staying safe.

Besides online RL, it is possible to learn from pre-collected data since usually large amounts of data have been collected with existing policies. However, certifying constraint satisfaction, including safety constraints, during off-policy evaluation in sequential decision making, is not straightforward and could be challenging. Batch policy learning under constraints [128] adapted abundant (non-optimal) behavior data to a new policy, with provable guarantees on constraint satisfaction. Authors have proposed an algorithmic framework for learning policies from off-policy data respecting both primary objective and constraint satisfaction. The framework has been successfully applied to OpenAI's CarRacing environment, where offline data was collected from 1500 trajectories of double-DQN's randomization. During learning, two behavioral constraints were respected: smooth driving and lane centering.

An algorithm for measuring the safety of RL agents has been proposed in [15]. A controller is modelled to characterize the actions taken by the agent and the possibilities that result from taking them. The controller is modelled as a Markov Decision Process, and probabilistic model checking techniques are leveraged to produce probability guarantees on the behaviour of the agent. The verification of the controller's model aims at finding the probability of reaching a failure state given a particular initial state. They have implemented their case study using OpenAI Gym environments. Results show that the authors were able to find safe probability bounds on the controller's behaviour. The model as well as the environment are very simple, thus the approach should be evaluated on more complex environments.

Using Control Barrier Functions

Barrier functions have been widely used to design safe RL-based controllers. Based on its definition, the value of a barrier function on a point is increased to infinity, as the point approaches boundaries of the feasible region of an optimization problem. Control Barrier Functions (CBF) plays a role equivalent to Lyapunov functions in the study of liveness properties of dynamical systems; if one finds a CBF for a given system, it becomes possible to define the set of admissible initial states and a feedback strategy that ensures safety for that system by indicating unsafe states. For example, [155] proposed a safe learning-based controller using CBF and actor-critic architecture. The authors have augmented the original performance function with a CBF candidate, to penalize actions that violate safety constraints. CBFs, with the initial condition belonging to a predefined set, guarantee that the states of the system will stay within that set by imposing proper condition on the trajectories of a nonlinear system, playing a role similar to that of Lyapunov functions in stability guarantee. A CBF candidate is defined to be positive within the predefined safe set and it reaches infinity at the boundary of that set. Hence, the control policy is designed to make the derivative of CBF negative in the vicinity of the boundary. It is not easy to compare the proposed approach to peer approaches for safety although it has been evaluated on a lane changing scenario as a challenging task in vehicle autonomy showing its effectiveness. The applicability of the approach looks limited, since the employed network for actor-critic is too small. Moreover, a known set of safe states is required for employing CBFs, which is not the case for many real-world problems. Similarly, RL-CBF framework [39] is a combination of a model-free RL-based controller, model-based controllers utilizing CBFs and real-time learning of the unknown system dynamics. Since CBF-based controllers can guarantee safety and also conduct RL learning by restricting the set of explorable policies, the goal is to ensure the safety during learning while benefiting from high performance of RL-based controllers. The safe RL-CBF has been successfully evaluated on an autonomous car-following scenario with wireless vehicle-to-vehicle communication performing more efficiently than other state-of-the-art algorithms while staying safe. However, a significant limiting assumption in their work is that a valid invariant safe set of the system needs to be specified. A new actor-critic-barrier structure for multiplayer safety-critical games (systems) has been introduced in [258]. Assuming that a system is represented by a non-zero-sum (NZS) game with full-state constraints, a barrier function was used to transform the game to an unconstrained NZS. An actor-critic model with deep RL architecture is then utilized to learn the Nash equilibrium in an online manner. Authors showed that the proposed structure, *i.e.* actor-critic-barrier, does not violate the constraints during learning, given that the initial state is in a predefined bound. A nonlinear system with two-players has been successfully used to evaluate the proposed structure along with analyzing its boundedness and stability. [51] have proposed an approach to train safe DNN controllers for cyber-physical systems so that they can satisfy given safety properties. The key idea of the approach is to embed safety properties into the RL. CBFs were employed to formulate a priori safety invariant property. The property is checked through an SMT-based verification process to impose penalties on undesired behaviors. To do so, if the verification results in failure, the RL cost function is modified using the generated counter-example to search towards safe policies. This process is repeated until achieving a proof of safety. As a proof of concept, three nonlinear dynamical system examples have been successfully tested.

- Adaptive approaches to safety should be considered for RL-based systems: Complying with pre-defined constraints is not enough and the system should be able to deal with situations that cannot be predicted. Similarly, assuming a prior set of safe states in many real-world problems is not realistic.
- Although existing results of certifying stability of RL-based controllers are interesting, the application scope is still non-practical: certification usually is performed for toy problems or simple control tasks, so they should be extended to address realistic environments.
- Using other ML approaches to make safe decisions in RL is not safe: outcomes of any ML-based systems should be investigated to be error-free and safe to apply, therefore one can not rely on their prediction for safety of RL agents.
- Generalizability of uncertainty estimation can be challenging: an approach may successfully estimate failure distribution for a particular situation but not for others. Therefore, various safety critical scenarios/environments should be tested to assess the effectiveness of such methods.

5.6 Direct Certification

By "Direct Certification" we refer to any paper that aims at a general framework for ML certification through a certain number of methods, considerations or steps. Although no ISO 26262 type standard has been created yet for ML-based systems, some papers indeed tackle this challenge. As such, those papers differ from papers described in previous sections, as they do not use a specific method to cover a certain aspect of certification. Instead, they try to address higher level considerations and attempt to describe how all those methods could fit together to reach a viable certification process.

Most papers rely on already established concepts such as; assurance cases type, graph structured notation (GSN) or even adaptation of standards. These concepts are viewed as a foundational starting point to be adapted to the specificities of ML. [108] used GSN with continuous engineering as a structure to take into consideration what they define as "uncertainties of ML" that is; imprecision of higher-level goals, lack of evidence and identification of feasible goals. The structured notation allows for a simple decomposition of those concepts, in order to have a simpler identification of the sub-concepts needed in the safety process. The GSN can also be used [64][192][69] to build an approach similar to assurance cases and hazard analysis. [64] defined multiple safety layers for verification, with for instance a layer about different states of the system, *e.g.* normal or emergency, with specification of what to do. If those methods guarantee a better comprehension of what is desired, they remain very high-level.

We identified some papers [194][195][101], that discussed how ISO 26262 could be adapted to ML. The authors mainly discuss the main obstacles for adaptation; notably the lack of specification that is not covered strictly by train data and the lack of interpretability of models. These obstacles are suggested to be tackled by using tolerance techniques, reducing the ML components to their simplest form, increasing the models interpretability, using artifact tracing in order to link data features to ML component output, and computing measures of uncertainty. The authors also advocate for coverage metrics/safety envelopes to investigate training data and recommend avoiding end-to-end approaches, that is a system solely based on ML. They argue that such a system would be incompatible with the ISO 26262 framework assumptions about stability of components, since the ML model weights change depending on the training and training set. Moreover, to circumvent the actual problem, it is recommended to use techniques based on intent/maturity rather than clear specification, which echoes the "Overarching Goals" we mentioned in Introduction. Note that some methods described in previous sections highlight the necessity to take into consideration aspects of ISO 26262, and even propose some techniques to serve as a basis for further inquiries of specific points of the standards. For instance, [96] covered safety constraints built through expert knowledge and/or statistical data with a given severity level, which can then be assessed for safety violation in simulation. This could serve as a good basis for further analysis or evaluation of safety requirements which is one point made by ISO 26262.

While ISO 26262 is the most studied proxy for ML certifications standard, it is not the only one. As discussed in the introduction, ISO21448 (SOTIF) intended functionality is an interesting standard for ML, as it focuses on unexpected behavior, such as prediction on OOD and Adversarial Examples. Aside from techniques mentioned in previous sections, [103] developed on the OOD problem and how it relates to SOTIF. [172] proposed an overall iterative generic (OGI) method for developing safe-by-design AI-based systems. This method relies upon a generic architecture and safety principles focusing on events indistinguishability, targets variability, noises, error propagation and human-machine interactions harmonization, which are concerns raised in SOTIF. For air transportation, [41] discussed a run-time assurance based on the ASTM F3269-17 for bounded behavior of complex systems. They mainly rely on monitoring components with backup functionality. They showed through experiment that with three monitors and a contingency mechanism, they limit ML component erroneous behavior without raising false alarms.

In [24] a different approach for the actual certification of ML was suggested; it relies on an architecture to control the ML component, with the fault recovery mechanism being the cornerstone. They claim that, using this architecture, DNN no longer needs to be certified, only the fault recovery system needs to be. This observation is similar to techniques using monitors to control the ML actuators such as [41] that was discussed previously or [117] which made use of ASIL (Automotive Safety Integrated Level). However, to be effective, this requires that the monitor can be more easily certified than the actual ML system. In all of those studies, little to none practical case study were presented with detailed use cases.

Other papers [28][27][9][24][70][177] raised similar observations about ML limits or issues, which we mentioned in the previous sections. Those issues represent a direct threat to certification; lack of specification, distributional shift, adversarial examples, out-of-distribution problems, lack of interpretability, lack of testing/verification approaches. Methods presented earlier are cited as being a way to tackle precise problems, but they also make the case for more traditional techniques such as; redundancy and fault tolerance subsystems, look ahead components, backup systems, coverage criteria or traceability through collection of Neural Networks related artifacts such as weights or versions. For instance, [70] proposed an approach where the requirements of a machine learning model drives the safety assurance process for the machine learning model. The process splits into 5 stages: requirements elicitation, data management, model learning, model verification and model deployment. Each stage aims at ensuring that the machine learning model is safe, when integrated in the real world.

Described techniques do not necessarily make use of all the considerations we elaborated on in previous sections, showing there is still ground for a unified process. As such, there is still no concrete standard or draft of standard. We have nonetheless illustrated that this preoccupation is currently understood by the scientific community.

- Some efforts have been made to adapt existing standards or considerations to specifics of ML, in particular ISO26262 seems a promising basis for car related systems.
- However, there is no clear process established and most studies offer incomplete approaches with no practical real case studies.
- Considering the plethora of methods developed within the various certification categories (Robustness, Uncertainty, Explainability, Verification), there are countless opportunities to apply them simultaneously in use case studies, inching closer to universal ML standards.

5.7 Others

This section regroups papers that couldn't fit in other categories and for which we give a brief overview of what they elaborate on.

- NN-dependability-kit [38] is a data driven toolbox that aims to provide directives to ensure uncertainty reduction in all the life cycle steps, notably robustness analysis with perturbation metrics and t-way coverage. This technique was used by [70] mentioned in the previous section with its requirements-driven safety assurance approach for ML. Building upon NN-dependability-kit, "specific requirements that are explicitly and traceably linked to system-level safety analysis" were tackled. While NN-dependability-kit did not focus on this aspect, it is an important one to comply with safety issues.

- Instead of trying to improve the model resilience through loss regularization or other mechanisms, some techniques focused on post training reparation of the model; the common idea is to search through the DNN for neurons that could lead to unexpected behavior and patch them. [209] used Particle Swarm Optimization (PSO) to modify weights of layers to correct faulty behavior. [73] instead used Markov Decision Process (MDP) to repair a model through safety properties that can directly be exploited to modify weights during training. They also show that with this technique they can "repair" data, by screening noisy data that would be outside of a safety envelope and remove those outliers.
- Generally used in object recognition, *bounding boxes* are a technique to increase safety. [36] applied it on 3d points cloud Neural Network PIXOR whose task is to predict the final position of an object. They split the decision part of the algorithm into non-critical and critical area detection, critical area symbolizing where the model should process carefully. For the latter part, they use a non-max-inclusion algorithm, which enlarges the prediction area of the same object by taking into account boxes with lower probability, in order to remain conservative as ground truth is not known in operation.
- DNN have been quite developed in our paper, as they represent a huge portion of the ML model used. However, there are some efforts to bring certification processes also for other models' types. [213] proposed a method for augmenting decision trees through expert knowledge with refinement to reduce the number of variables. They further leverage the information gain metric to "prune" the decision tree to retain an optimal decision tree. However, they assume attributes are independent, so a combination can exist in the tree without really existing in actual possibilities which can induce biases.
- Another interesting addition to safety-critical certification could come from transfer learning. Transfer learning is a learning technique which aims at reusing a part or an entire pre-trained model on task A, in order to adapt it to task B. This allows for decreased training time as well as potentially increased accuracy. This approach could be useful in safety-critical applications when environment modifications happen. Indeed, every time a sensor configuration is modified, one would like to avoid retraining the network from scratch to account for this modification. Transferring already established knowledge could help solve this issue. [198] proposed an approach based on neural networks to calculate a transformation matrix mapping each input from an existing domain to the new one, helping at the same time to understand how data are mapped between domains. They mainly tested it on a lane-changing driving problem. However, further studies would be required to take into consideration harmful consequences that transferred knowledge could bring.

- [137] deals with the person re-identification problem, *i.e.* matching pedestrian images observed from multiple non-overlapping cameras. Traditional feature learning methods (extracting discriminating features and matching them in between images) are limited due to the various viewing conditions on real images. The approach proposed is based on dictionary learning, that tries to reconstruct sample images using a dictionary of features and some sparse coefficients. They extend it to be able to handle two-view. They try to learn a dictionary and a projection of the two different views, so that the reconstruction of the sample images (with the dictionary and projection) is close to the original images, as well as minimizing the intersection of the two views via regularization.
- [212] tackles data poisoning attack, *i.e.* a game between a defender and an attacker. The goal of the defender is to learn a good model, while the attacker's goal is for the defender to fail. N data points are selected, the attacker choose M poisoned data points such as $M := \lfloor \epsilon N \rfloor \leq N$ (depending the budget ϵ of attacker), the defender learns on the $M + N$ data points and aim to minimize the loss. The poisoned data D_p consists only of new data, that is the attacker cannot modify existing sane data D_c. The authors use a modified version of data sanitized defense, consisting of finding the poisoned data and removing it. They upper bound the worst test-case loss under attack with $\max_{D_p \subseteq \mathcal{F}} \min_{\theta} \mathcal{L}(D_p \cap D_c)$, where $\mathcal{F}$ is called the feasible set, and can be for instance a neighborhood sphere of a given radius. The main assumption of the method is that all clean data points are feasible *e.g.* $D_c \cap \mathcal{F} \neq \emptyset$.
- Statistical metrics can be leveraged to improve safety of ML-based systems. [13] does so by first training an offline model with a trusted dataset, gathering information such as cumulative distribution of classes data, accuracy of model, etc. The system can then be put online and can compare new influx of data through the same statistics using confidence level. If a high divergence is measured, the system can report the error.
- [132] proposes an approach for a safe visual-based navigation system by exploiting perceptual control policies. To that end, a model predictive network, which itself relies on a model predictive controller, is used to provide information about the vehicle and select regions of interest on the visual input. This information is considered as expert trajectories and is used through imitation learning to learn a perceptual controller of the navigation system. The perceptual controller, which is the main contribution of this approach outperforms baselines, by quickly detecting unsafe conditions that the navigation system might encounter through uncertainty quantification.
- Finally, [125] describes the use of safety cages to control actions in an autonomous vehicle. Safety cages are generally used on black box systems where we do not have full understanding of how the system works. They limit unsafe actions the system can take. Their approach is based on Imitation Learning, which learns from a simulation based autonomous vehicle.

From the information collected via imitation learning, the safety process leads the autonomous vehicle to avoid collisions.

6 Future Direction and Research Opportunities in ML Certification

The low-level, technical considerations we derived from our paper reviews allowed us to present state-of-the-art techniques as well as existing challenges and limitations. We now further discuss on a higher level what are the future possibilities for ML certifications in accordance with what we discussed earlier.

Increasing diversity of safety-critical use cases

Safety-critical considerations in ML have been gaining a lot of attention over the past years; since 2017, the number of papers dealing with this topic has been steadily rising, which shows the growing preoccupation of the community on this topic. Concerning "certification" considerations, we find that most research is about applications involving image data, which could be explained by the fact that it is easily accessible and visually interpretable. As such, the majority of the datasets used in experiments were: MNIST, CIFAR and ImageNet. While these datasets are interesting to introduce new concepts and theory, in particular with regards to understanding models behaviors, their relevance seems quite limited, when it comes to empirically evaluating techniques for safety-critical systems in aviation or automotive. Only about a fifth of all datasets used are directly related to such fields. Moreover, while aviation seems to have a standard dataset with ACAS-XU, there is no clear "driving" dataset that is widespread in the case of automotive. Moreover, even though the goal is to approach all of ML from a safety-critical point of view, Deep Neural Networks (especially Convolutional Neural Networks) are overwhelmingly represented. We suspect this is because DNNs are amongst the most versatile models available, as they can handle multiple data types and tasks. However, it must be noted that DNNs are also harder to train and interpret, compared to more traditional ML algorithms like Random Forests or Support-Vector Machines. We argue that these simpler models would also benefit from the safety-considerations presented in the previous section.

Bolstering partnership between Academia & Industries

While academia research represents the majority of the papers we screened in our study (around 60%), we believe that collaborations between academia and industry (only around 32% of screened papers) represent a great opportunity for the research on ML certification. Indeed, as explained in the review, assessing the critical-safety of ML systems requires more practical datasets, which could be accessed through industrial partnership. A deeper collaboration between industry and academia would be mutually beneficial as it would encourage researchers to adapt the current ML models (or develop new ones) to meet the specific constraints of a given industrial partner. In the short term, we believe that adapting ML methods to respect industrial constraints on a partner-by-partner basis is more realistic than aiming at developing uniform safety standards. We therefore think that initiatives such as DEEL[24] (DEpendable and Explainable Learning) which allows for cooperation between academia and industry are to be encouraged, in order to foster the development of suitable concepts and techniques.

Adapting proven techniques to ML specificities

ML certification research should not be restricted to devising new techniques, as already established methods can be useful, while already benefiting from a solid background. The verification techniques we presented are perfect examples. Whether it is using Formal Solver such as Linear Programming or more classical Software Engineering techniques such as MC/DC, there are already plenty of existing techniques that could potentially offer extra safety to ML. Those techniques would only require adaptation to the specificity that ML brought to the table. For instance, [216] devised a criteria adapted to DNNs that is analogous to the traditional MC/DC, which is used in classical software testing.

Deriving formal guarantees

Currently, only the robustness property of ML models seems to have been rigorously formalized in some cases, therefore yielding certain theoretical guarantees which are important to supplement empirical evidence. Indeed, empirical evidence is limited by the fact that it can only assess the safety of the model on the finite datasets used in experiments. Other certification sub-fields such as OOD detection, Uncertainty, Explainability, and Testing currently lack formalism and guarantees. For example, in OOD detection, no formal definition of what constitute the set of all OOD inputs is used and OOD detectors are currently evaluated by simply measuring their ability to discriminate between instances from the dataset used for training (MNIST for example), and instances taken from a completely different OOD dataset (notMNIST for example). Although this type of experiment is helpful to compare OOD detectors, it cannot provide true guarantees that the model will perform safely, when put into production as the OOD dataset may not be representative of all possible OOD inputs. Moreover, we believe that the applicability of these techniques requires guarantees on False Negative rates, *i.e.* how often an unsafe prediction on a OOD sample will be performed in deployment, without raising an alarm, which are yet to be developed.

Finding new avenues to complement formal guarantees

It is possible that formal guarantees might not be achievable for certain parts of the ML framework. This could potentially be true for sub-fields of certification where one does not have access to ground-truth values (oracle problem). For uncertainty, although engineers can have a high-level intuition of what uncertainty is, it is not clear whether or not the specific computations of aleatoric

[24] `https://www.deel.ai`

and epistemic uncertainties are meaningful. There is a high-level notion that good uncertainties should correlate with model error, although there is no universal way to evaluate the adequacy of the uncertainty estimation as a proxy for the prediction failure. In post-hoc explainability, it is not always possible to know what is the ground-truth for explanations, because the models that are studied are black box by nature and because it is hard to extract human-understandable summaries of the reasoning behind their decisions. When testing data quality, it is not clear what qualifies as “good data” and what metrics can encode the right notions of quality. This specific issue faced in testing is in a sense similar to the non-testable program paradigm introduced by [248]. Even formal verification is limited by the formalization of the property it aims to verify. If a property cannot be properly formulated, it cannot be verified. Taking inspiration from differential testing, the lack of ground truth could be tackled by introducing novel pseudo-oracles. This would in turn possibly lead to other forms of guarantees.

Studying cross sub-fields of certifications

Most papers focus on one category in search of interesting results. However, in practice, all the sub-fields are expected to work hand in hand. As such, more studies should focus on studying the connections between different sub-fields, the potential trade-offs involved, and what techniques can negatively impact each other. For instance, [200] demonstrated that adversarial OOD can fool both OOD and AE dedicated detectors. Possible connections we identified in our review are the following:

– Robustness is linked to OOD detection as the right OOD detector should reject all samples on which the system cannot be trusted while Robustness measures in a sense "how much" samples we can safely predict on. More generally, the notion of distributional shift, that encompasses those two sub-fields, could benefit from a unified treatment.
– We suspect that Robust training and Explainability of DNNs are deeply connected. One the one hand, we could expect adversarially robust neural networks to have more interpretable saliency maps because their decisions cannot be significantly altered by spurious perturbations of the input. Therefore saliency maps could potentially highlight more human-readable features from the input images. On the other hand, as noted in [151], to reliably defend against adversarial attacks, DNNs require more capacity, making them less interpretable in the process. Hence, connections between model capacity, robustness, and explainability are not trivial and should be explored more thoroughly.
– Uncertainty and Explainability share a lot of similarities. Indeed, they both attempt to help machines mimic how human beings make decisions, therefore increasing the trust users have in models. For this reason, we think that further studying the relation between model uncertainty and explainability could foster understanding in the two respective domains. For instance, we suggest adapting some of the metrics used in uncertainty to explain-

ability. As stated previously, uncertainties are currently used as proxies for prediction confidence, and are expected to be high on instances where the model fails and low on instances where it predicts correctly. Similar notions could be extended to post-hoc explanations, *e.g.* when studying an instance on which the model fails to make the right prediction, we would like the explanation to be aberrant or misleading. For example, it could possibly be observed that during miss-classification of an image, the network is putting a lot of its attention on irrelevant features, such as the background. Measuring how much specific explanation methods can help prevent miss-classifications could provide promising quality metrics.
– There are also connections to be made between Adversarial Robustness and Uncertainty. As discussed earlier, there are techniques from both categories that modify the standard training procedure by adding adversarial loss (Robustness), loss attenuation (Aleatoric Uncertainty), and MCDropout (Epistemic Uncertainty). It is unclear to us how all these different training objectives would interact if they are all used in tandem. Although loss attenuation and MCDropout have previously been used simultaneously during training [119], it seems that the further addition of an adversarial loss remains to be investigated.
– Verification and Explainability/Uncertainty. Most work on verification, whether it is through formal methods or test-based generation, focuses on generating adversarial examples or verifying robustness properties. However, we would like to point out that such approaches could also be useful to verify equivalent properties for other concepts related to explainability or uncertainty through formal checking (although a more formal framework for both would be required), or generation of "corner-case", as it is currently done for robustness to adversarial examples.
– Data quality and all other fields. Many techniques we have reviewed focus on certifying properties of a fixed model. While there exists a relation between data and models (since the former is used to train the latter), it seems to us that there are some inherent limitations in focusing only on models. In particular, a model can behave correctly regarding one of the properties, while the data itself does not represent the full picture, hence possibly leading to error when the model is put in operation conditions. As such, methods which focus on studying how such considerations are present *directly* in the data are important and why recent efforts have been focused on bettering their collection, processing and analysis [189].
– Finally, we observed studies in the literature that apply Robustness and/or Uncertainty techniques to increase the safety of Reinforcement Learning agents. It would be interesting to go a step further and add explainability to the picture. Indeed, getting insight on the decision-making process of an agent would be an important step in making sure that no future decisions are unsafe. More precisely, when the state of the environment is fed to the agent as an image, saliency maps of employed DNNs, for example Q-network in the DQN, could be computed to help diagnose faulty behaviors of the agent.

Those are just to name a few possible connections that exist between the various certification sub-fields. Overall, we believe that studying certification of ML from multiple aspects at the same time would not only bring more knowledge into each given sub-field, but would help make significant steps toward a global certification approach. We suspect that, as modules implementing techniques from all the sub-fields become readily available in popular Deep Learning frameworks such as TensorFlow and Pytorch, we will see an increase in cross sub-fields studies. As an example, robustness modules for Tensorflow/Keras are now available [14].

Unifying all sub-fields for a complete standard

As already mentioned, there is currently no standard for ML certification. The attempts that were discussed in **Section 5.6** generally stick to high-level considerations, do not cover all aspects of a system certification, and/or do not take into account possible trade-off of using different certifications aspects at the same time. However, it is clear that there is some basis that can be adapted, using previously defined standards such as ISO 26262 to name only the most studied one. We believe that a true standard can only be defined if the expertise from all the discussed sub-fields is brought together. In this perspective, a long term goal for the research community would be to devise a framework that provides clear-cut criteria, to generate models that are robust to adversarial perturbations and distributional shift (Robustness), that know what they don't know (Uncertainty and OOD detection), that can provide insight on how their decisions are made (Explainability) and whose properties can be verified/tested (Verification).

7 Conclusion

This paper provides a comprehensive overview of certification challenges for ML based safety-critical systems. We conducted a systematic review of the literature pertaining to *Robustness*, *Uncertainty*, *Explainability*, *Verification*, *Safe Reinforcement Learning* and *Direct Certification*. We identified gaps in this literature and discussed about current limitations and future research opportunities. With this paper, we hope to provide the research community with a full view of certification challenges and stimulate more collaborations between academia and industry.

Acknowledgements We would like to thank the following authors who kindly provided us feedback about our review of their work: Mahum Naseer, Hoang-Dung Tran, Jie Ren, David Isele, Jesse Zhang, Michaela Klauck, Guy Katz, Patrick Hart, Guy Amit, Yu Li, Anurag Arnab, Tiago Marques, Taylor T. Johnson, Molly O'Brien, Kimin Lee, Lukas Heinzmann, Björn Lütjens, Brendon G. Anderson, Marta Kwiatkowska, Patricia Pauli, Anna Monreale, Alexander Amini, Joerg Wagner, Adrian Schwaiger, Aman Sinha, Joel Dapello, Kim Peter Wabersich. Their feedback contributed to improving this SLR.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Abreu S (2019) Automated architecture design for deep neural networks. arXiv preprint arXiv:190810714
2. Agostinelli F, Hocquet G, Singh S, Baldi P (2018) From reinforcement learning to deep reinforcement learning: An overview. In: Braverman Readings in Machine Learning. Key Ideas From Inception to Current State, Springer, pp 298–328
3. Alagöz I, Herpel T, German R (2017) A selection method for black box regression testing with a statistically defined quality level. In: 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST), pp 114–125, DOI 10.1109/ICST.2017.18
4. Amarasinghe K, Manic M (2019) Explaining what a neural network has learned: Toward transparent classification. In: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, pp 1–6
5. Ameyaw DA, Deng Q, Söffker D (2019) Probability of detection (pod)-based metric for evaluation of classifiers used in driving behavior prediction
6. Amini A, Schwarting W, Soleimany A, Rus D (2019) Deep evidential regression. arXiv preprint arXiv:191002600
7. Amit G, Levy M, Rosenberg I, Shabtai A, Elovici Y (2020) Glod: Gaussian likelihood out of distribution detector. URL `https://arxiv.org/abs/2008.06856v3`, `2008.06856`
8. Anderson BG, Ma Z, Li J, Sojoudi S (2020) Tightened convex relaxations for neural network robustness certification. In: 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, pp 2190–2197
9. Aravantinos V, Diehl F (2019) Traceability of deep neural networks. `1812.06744`
10. Arcaini P, Bombarda A, Bonfanti S, Gargantini A (2020) Dealing with robustness of convolutional neural networks for image classification. In: 2020 IEEE International Conference On Artificial Intelligence Testing (AITest), pp 7–14, DOI 10.1109/AITEST49225.2020.00009
11. Arnab A, Miksik O, Torr PHS (2018) On the robustness of semantic segmentation models to adversarial attacks. `1711.09856`
12. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R, et al. (2020) Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion 58:82–115
13. Aslansefat K, Sorokos I, Whiting D, Kolagari RT, Papadopoulos Y (2020) Safeml: Safety monitoring of machine learning classifiers through statistical difference measure. `2005.13166`

14. Ayers EW, Eiras F, Hawasly M, Whiteside I (2020) Parot: a practical framework for robust deep neural network training. In: NASA Formal Methods Symposium, Springer, pp 63–84
15. Bacci E, Parker D (2020) Probabilistic guarantees for safe deep reinforcement learning. In: International Conference on Formal Modeling and Analysis of Timed Systems, Springer, pp 231–248
16. Baheri A, Nageshrao S, Tseng HE, Kolmanovsky I, Girard A, Filev D (2019) Deep reinforcement learning with enhanced safety for autonomous highway driving. In: 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, pp 1550–1555
17. Bakhti Y, Fezza SA, Hamidouche W, Déforges O (2019) Ddsa: a defense against adversarial attacks using deep denoising sparse autoencoder. IEEE Access 7:160397–160407
18. Baluta T, Shen S, Shinde S, Meel KS, Saxena P (2019) Quantitative verification of neural networks and its security applications. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp 1249–1264
19. Bar A, Huger F, Schlicht P, Fingscheidt T (2019) On the robustness of redundant teacher-student frameworks for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 0–0
20. Bar A, Klingner M, Varghese S, Huger F, Schlicht P, Fingscheidt T (2020) Robust semantic segmentation by redundant networks with a layer-specific loss contribution and majority vote. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 332–333
21. Ben Braiek H, Khomh F (2019) Deepevolution: A search-based testing approach for deep neural networks. In: 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), pp 454–458, DOI 10.1109/ICSME.2019.00078
22. Berkenkamp F, Turchetta M, Schoellig AP, Krause A (2017) Safe model-based reinforcement learning with stability guarantees. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), p 908–919
23. Bernhard J, Gieselmann R, Esterle K, Knol A (2018) Experience-based heuristic search: Robust motion planning with deep q-learning. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp 3175–3182
24. Biondi A, Nesti F, Cicero G, Casini D, Buttazzo G (2020) A safe, secure, and predictable software architecture for deep learning in safety-critical systems. IEEE Embedded Systems Letters 12(3):78–82, DOI 10.1109/LES.2019.2953253
25. Bragg J, Habli I (2018) What is acceptably safe for reinforcement learning? In: International Conference on Computer Safety, Reliability, and Security, Springer, pp 418–430

26. Bunel R, Mudigonda P, Turkaslan I, Torr P, Lu J, Kohli P (2020) Branch and bound for piecewise linear neural network verification. Journal of Machine Learning Research 21(2020)
27. Burton S, Gauerhof L, Heinzemann C (2017) Making the case for safety of machine learning in highly automated driving. In: Tonetta S, Schoitsch E, Bitsch F (eds) Computer Safety, Reliability, and Security, Springer International Publishing, Cham, pp 5–16
28. Burton S, Gauerhof L, Sethy BB, Habli I, Hawkins R (2019) Confidence arguments for evidence of performance in machine learning for highly automated driving functions. In: Romanovsky A, Troubitsyna E, Gashi I, Schoitsch E, Bitsch F (eds) Computer Safety, Reliability, and Security, Springer International Publishing, Cham, pp 365–377
29. Cardelli L, Kwiatkowska M, Laurenti L, Patane A (2019) Robustness guarantees for bayesian inference with gaussian processes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 7759–7768
30. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. `1608.04644`
31. Castelvecchi D (2016) Can we open the black box of ai? Nature News 538(7623):20
32. Chakrabarty A, Quirynen R, Danielson C, Gao W (2019) Approximate dynamic programming for linear systems with state and input constraints. In: 2019 18th European Control Conference (ECC), IEEE, pp 524–529
33. Chen TY, Cheung SC, Yiu SM (2020) Metamorphic testing: a new approach for generating next test cases. arXiv preprint arXiv:200212543
34. Chen Z, Narayanan N, Fang B, Li G, Pattabiraman K, DeBardeleben N (2020) Tensorfi: A flexible fault injection framework for tensorflow applications. `2004.01743`
35. Cheng C, Nührenberg G, Yasuoka H (2019) Runtime monitoring neuron activation patterns. In: 2019 Design, Automation Test in Europe Conference Exhibition (DATE), pp 300–303, DOI 10.23919/DATE.2019.8714971
36. Cheng CH (2020) Safety-aware hardening of 3d object detection neural network systems. `2003.11242`
37. Cheng CH, Nührenberg G, Ruess H (2017) Maximum resilience of artificial neural networks. In: International Symposium on Automated Technology for Verification and Analysis, Springer, pp 251–268
38. Cheng CH, Huang CH, Nührenberg G (2019) nn-dependability-kit: Engineering neural networks for safety-critical autonomous driving systems. `1811.06746`
39. Cheng R, Orosz G, Murray RM, Burdick JW (2019) End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 3387–3395
40. Cobbe K, Klimov O, Hesse C, Kim T, Schulman J (2019) Quantifying generalization in reinforcement learning. In: International Conference on

Machine Learning, PMLR, pp 1282–1289
41. Cofer D, Amundson I, Sattigeri R, Passi A, Boggs C, Smith E, Gilham L, Byun T, Rayadurgam S (2020) Run-time assurance for learning-based aircraft taxiing. In: 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), pp 1–9, DOI 10.1109/DASC50938.2020.9256581
42. Colangelo F, Neri A, Battisti F (2019) Countering adversarial examples by means of steganographic attacks. In: 2019 8th European Workshop on Visual Information Processing (EUVIP), pp 193–198, DOI 10.1109/EUVIP47703.2019.8946254
43. Cosentino J, Zaiter F, Pei D, Zhu J (2019) The search for sparse, robust neural networks. 1912.02386
44. Croce F, Hein M (2019) Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. arXiv preprint arXiv:190511213
45. Croce F, Andriushchenko M, Hein M (2019) Provable robustness of relu networks via maximization of linear regions. In: the 22nd International Conference on Artificial Intelligence and Statistics, PMLR, pp 2057–2066
46. Daniels ZA, Metaxas D (2018) Scenarionet: An interpretable data-driven model for scene understanding. In: IJCAI Workshop on Explainable Artificial Intelligence (XAI) 2018
47. Dapello J, Marques T, Schrimpf M, Geiger F, Cox DD, DiCarlo JJ (2020) Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. BioRxiv
48. Dean S, Matni N, Recht B, Ye V (2020) Robust guarantees for perception-based control. In: Learning for Dynamics and Control, PMLR, pp 350–360
49. Delseny H, Gabreau C, Gauffriau A, Beaudouin B, Ponsolle L, Alecu L, Bonnin H, Beltran B, Duchel D, Ginestet JB, Hervieu A, Martinez G, Pasquet S, Delmas K, Pagetti C, Gabriel JM, Chapdelaine C, Picard S, Damour M, Cappi C, Gardès L, Grancey FD, Jenn E, Lefevre B, Flandin G, Gerchinovitz S, Mamalet F, Albore A (2021) White paper machine learning in certified systems. 2103.10529
50. Demir S, Eniser HF, Sen A (2019) Deepsmartfuzzer: Reward guided test generation for deep learning. 1911.10621
51. Deshmukh JV, Kapinski JP, Yamaguchi T, Prokhorov D (2019) Learning deep neural network controllers for dynamical systems with safety guarantees. In: 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), IEEE, pp 1–7
52. Dreossi T, Ghosh S, Sangiovanni-Vincentelli A, Seshia SA (2017) Systematic testing of convolutional neural networks for autonomous driving. 1708.03309
53. Duddu V, Rao DV, Balas VE (2019) Adversarial fault tolerant training for deep neural networks. arXiv preprint arXiv:190703103
54. Dutta S, Jha S, Sanakaranarayanan S, Tiwari A (2017) Output range analysis for deep neural networks. arXiv preprint arXiv:170909130
55. Dybå T, Dingsøyr T (2008) Empirical studies of agile software development: A systematic review. Information and Software Tech-

nology 50(9):833 – 859, DOI https://doi.org/10.1016/j.infsof.2008.01.006, URL http://www.sciencedirect.com/science/article/pii/S0950584908000256
56. Eniser HF, Gerasimou S, Sen A (2019) Deepfault: Fault localization for deep neural networks. 1902.05974
57. Everett M, Lutjens B, How JP (2020) Certified adversarial robustness for deep reinforcement learning. arXiv preprint arXiv:200406496
58. Fan DD, Nguyen J, Thakker R, Alatur N, Agha-mohammadi Aa, Theodorou EA (2020) Bayesian learning-based adaptive control for safety critical systems. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp 4093–4099, DOI 10.1109/ICRA40945.2020.9196709
59. Feng D, Rosenbaum L, Glaeser C, Timm F, Dietmayer K (2019) Can we trust you? on calibration of a probabilistic object detector for autonomous driving. arXiv preprint arXiv:190912358
60. Feng Y, Shi Q, Gao X, Wan J, Fang C, Chen Z (2020) Deepgini: Prioritizing massive tests to enhance the robustness of deep neural networks. 1903.00661
61. Fisac JF, Akametalu AK, Zeilinger MN, Kaynama S, Gillula J, Tomlin CJ (2018) A general safety framework for learning-based control in uncertain robotic systems. IEEE Transactions on Automatic Control 64(7):2737–2752
62. François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J (2018) An introduction to deep reinforcement learning. Foundations and Trends® in Machine Learning 11(3-4):219–354, DOI 10.1561/2200000071, URL http://dx.doi.org/10.1561/2200000071
63. Fremont DJ, Chiu J, Margineantu DD, Osipychev D, Seshia SA (2020) Formal analysis and redesign of a neural network-based aircraft taxiing system with verifai. In: International Conference on Computer Aided Verification, Springer, pp 122–134
64. Fujino H, Kobayashi N, Shirasaka S (2019) Safety assurance case description method for systems incorporating off-operational machine learning and safety device. INCOSE International Symposium 29(S1):152–164, DOI https://doi.org/10.1002/j.2334-5837.2019.00676.x, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2334-5837.2019.00676.x, https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2334-5837.2019.00676.x
65. Fulton N, Platzer A (2019) Verifiably safe off-model reinforcement learning. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, pp 413–430
66. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. 1506.02142
67. Gambi A, Mueller M, Fraser G (2019) Automatically testing self-driving cars with search-based procedural content generation. In: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, Association for Computing Machinery, New York,

NY, USA, ISSTA 2019, p 318–328, DOI 10.1145/3293882.3330566, URL https://doi.org/10.1145/3293882.3330566
68. Gandhi D, Pinto L, Gupta A (2017) Learning to fly by crashing. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 3948–3955
69. Gauerhof L, Munk P, Burton S (2018) Structuring validation targets of a machine learning function applied to automated driving. In: Gallina B, Skavhaug A, Bitsch F (eds) Computer Safety, Reliability, and Security, Springer International Publishing, Cham, pp 45–58
70. Gauerhof L, Hawkins R, Picardi C, Paterson C, Hagiwara Y, Habli I (2020) Assuring the safety of machine learning for pedestrian detection at crossings. In: Casimiro A, Ortmeier F, Bitsch F, Ferreira P (eds) Computer Safety, Reliability, and Security, Springer International Publishing, Cham, pp 197–212
71. Gehr T, Mirman M, Drachsler-Cohen D, Tsankov P, Chaudhuri S, Vechev M (2018) Ai2: Safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE Symposium on Security and Privacy (SP), IEEE, pp 3–18
72. Ghosh S, Berkenkamp F, Ranade G, Qadeer S, Kapoor A (2018) Verifying controllers against adversarial examples with bayesian optimization. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 7306–7313
73. Ghosh S, Jha S, Tiwari A, Lincoln P, Zhu X (2018) Model, data and reward repair: Trusted machine learning for markov decision processes. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), pp 194–199, DOI 10.1109/DSN-W.2018.00064
74. Gladisch C, Heinzemann C, Herrmann M, Woehrle M (2020) Leveraging combinatorial testing for safety-critical computer vision datasets. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 1314–1321, DOI 10.1109/CVPRW50498.2020.00170
75. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, http://www.deeplearningbook.org
76. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:14126572
77. Goodman B, Flaxman S (2017) European union regulations on algorithmic decision-making and a "right to explanation". AI magazine 38(3):50–57
78. Göpfert JP, Hammer B, Wersing H (2018) Mitigating concept drift via rejection. In: International Conference on Artificial Neural Networks, Springer, pp 456–467
79. Gopinath D, Katz G, Pasareanu CS, Barrett C (2017) Deepsafe: A data-driven approach for checking adversarial robustness in neural networks. arXiv preprint arXiv:171000486

80. Grefenstette E, Stanforth R, O'Donoghue B, Uesato J, Swirszcz G, Kohli P (2018) Strength in numbers: Trading-off robustness and computation via adversarially-trained ensembles. arXiv preprint arXiv:181109300
81. Grefenstette E, Stanforth R, O'Donoghue B, Uesato J, Swirszcz G, Kohli P (2018) Strength in numbers: Trading-off robustness and computation via adversarially-trained ensembles. 1811.09300
82. Gros S, Zanon M, Bemporad A (2020) Safe reinforcement learning via projection on a safe set: How to achieve optimality? arXiv preprint arXiv:200400915
83. Gros TP, Hermanns H, Hoffmann J, Klauck M, Steinmetz M (2020) Deep statistical model checking. In: International Conference on Formal Techniques for Distributed Objects, Components, and Systems, Springer, pp 96–114
84. Gschossmann A, Jobst S, Mottok J, Bierl R (2019) A measure of confidence of artificial neural network classifiers. In: ARCS Workshop 2019; 32nd International Conference on Architecture of Computing Systems, pp 1–5
85. 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
86. Gualo F, Rodriguez M, Verdugo J, Caballero I, Piattini M (2021) Data quality certification using iso/iec 25012: Industrial experiences. Journal of Systems and Software 176:110938, DOI https://doi.org/10.1016/j.jss.2021.110938, URL https://www.sciencedirect.com/science/article/pii/S0164121221000352
87. Guidotti D, Leofante F, Castellini C, Tacchella A (2019) Repairing learned controllers with convex optimization: a case study. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, pp 364–373
88. Guidotti R, Monreale A, Giannotti F, Pedreschi D, Ruggieri S, Turini F (2019) Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems 34(6):14–23
89. Guo J, Jiang Y, Zhao Y, Chen Q, Sun J (2018) Dlfuzz: differential fuzzing testing of deep learning systems. Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering DOI 10.1145/3236024.3264835, URL http://dx.doi.org/10.1145/3236024.3264835
90. Guo W, Mu D, Xu J, Su P, Wang G, Xing X (2018) Lemna: Explaining deep learning based security applications. In: proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 364–379
91. Hahn EM, Perez M, Schewe S, Somenzi F, Trivedi A, Wojtczak D (2019) Omega-regular objectives in model-free reinforcement learning. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, pp 395–412

92. Hart P, Rychly L, Knoll A (2019) Lane-merging using policy-based reinforcement learning and post-optimization. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, pp 3176–3181
93. Hasanbeig M, Kroening D, Abate A (2020) Towards verifiable and safe model-free reinforcement learning. CEUR Workshop Proceedings
94. Hein M, Andriushchenko M (2017) Formal guarantees on the robustness of a classifier against adversarial manipulation. arXiv preprint arXiv:170508475
95. Hein M, Andriushchenko M, Bitterwolf J (2019) Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
96. Heinzmann L, Shafaei S, Osman MH, Segler C, Knoll A (2019) A framework for safety violation identification and assessment in autonomous driving. In: AISafety@IJCAI
97. Hendrycks D, Dietterich T (2019) Benchmarking neural network robustness to common corruptions and perturbations. 1903.12261
98. Hendrycks D, Gimpel K (2018) A baseline for detecting misclassified and out-of-distribution examples in neural networks. 1610.02136
99. Hendrycks D, Basart S, Mazeika M, Mostajabi M, Steinhardt J, Song D (2020) Scaling out-of-distribution detection for real-world settings. 1911.11132
100. Henne M, Schwaiger A, Roscher K, Weiss G (2020) Benchmarking uncertainty estimation methods for deep learning with safety-related metrics. In: SafeAI@ AAAI, pp 83–90
101. Henriksson J, Borg M, Englund C (2018) Automotive safety and machine learning: Initial results from a study on how to adapt the iso 26262 safety standard. In: 2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS), pp 47–49
102. Henriksson J, Berger C, Borg M, Tornberg L, Englund C, Sathyamoorthy SR, Ursing S (2019) Towards structured evaluation of deep neural network supervisors. In: 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), pp 27–34, DOI 10.1109/AITest.2019.00-12
103. Henriksson J, Berger C, Borg M, Tornberg L, Sathyamoorthy SR, Englund C (2019) Performance analysis of out-of-distribution detection on various trained neural networks. In: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp 113–120, DOI 10.1109/SEAA.2019.00026
104. Huang X, Kwiatkowska M, Wang S, Wu M (2017) Safety verification of deep neural networks. In: International conference on computer aided verification, Springer, pp 3–29
105. Ignatiev A, Pereira F, Narodytska N, Marques-Silva J (2018) A sat-based approach to learn explainable decision sets. In: International Joint Conference on Automated Reasoning, Springer, pp 627–645
106. Inouye DI, Leqi L, Kim JS, Aragam B, Ravikumar P (2019) Diagnostic curves for black box models

107. Isele D, Nakhaei A, Fujimura K (2018) Safe reinforcement learning on autonomous vehicles. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 1–6
108. Ishikawa F, Matsuno Y (2018) Continuous argument engineering: Tackling uncertainty in machine learning based systems. In: Gallina B, Skavhaug A, Schoitsch E, Bitsch F (eds) Computer Safety, Reliability, and Security, Springer International Publishing, Cham, pp 14–21
109. ISO 26262 (2018) ISO 26262: Road vehicles – Functional safety. International Organization of Standardization
110. Jain D, Anumasa S, Srijith P (2020) Decision making under uncertainty with convolutional deep gaussian processes. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp 143–151
111. Jeddi A, Shafiee MJ, Karg M, Scharfenberger C, Wong A (2020) Learn2perturb: an end-to-end feature perturbation learning to improve adversarial robustness. `2003.01090`
112. Jin W, Ma Y, Liu X, Tang X, Wang S, Tang J (2020) Graph structure learning for robust graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA, KDD '20, p 66–74, DOI 10.1145/3394486.3403049, URL `https://doi.org/10.1145/3394486.3403049`
113. Julian KD, Kochenderfer MJ (2019) Guaranteeing safety for neural network-based aircraft collision avoidance systems. In: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), IEEE, pp 1–10
114. Julian KD, Sharma S, Jeannin JB, Kochenderfer MJ (2019) Verifying aircraft collision avoidance neural networks through linear approximations of safe regions. arXiv preprint arXiv:190300762
115. Julian KD, Lee R, Kochenderfer MJ (2020) Validation of image-based neural network controllers through adaptive stress testing. `2003.02381`
116. Kandel A, Moura SJ (2020) Safe zero-shot model-based learning and control: A wasserstein distributionally robust approach. arXiv preprint arXiv:200400759
117. Kaprocki N, Velikić G, Teslić N, Krunić M (2019) Multiunit automotive perception framework: Synergy between ai and deterministic processing. In: 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), pp 257–260, DOI 10.1109/ICCE-Berlin47944.2019.8966168
118. Katz G, Barrett C, Dill DL, Julian K, Kochenderfer MJ (2017) Reluplex: An efficient smt solver for verifying deep neural networks. In: International Conference on Computer Aided Verification, Springer, pp 97–117
119. Kendall A, Gal Y (2017) What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems, Curran Associates, Inc., vol 30
120. Kenton Z, Filos A, Evans O, Gal Y (2019) Generalizing from a few environments in safety-critical reinforcement learning. arXiv preprint arXiv:190701475

121. Key TFFLA (????) Finding invariants in deep neural networks
122. Kitchenham B (2004) Procedures for performing systematic reviews. Keele, UK, Keele University 33(2004):1–26
123. Kläs M, Sembach L (2019) Uncertainty wrappers for data-driven models. In: International Conference on Computer Safety, Reliability, and Security, Springer, pp 358–364
124. Kuppers F, Kronenberger J, Shantia A, Haselhoff A (2020) Multivariate confidence calibration for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 326–327
125. Kuutti S, Bowden R, Joshi H, de Temple R, Fallah S (2019) Safe deep neural network-driven autonomous vehicles using software safety cages. In: Yin H, Camacho D, Tino P, Tallón-Ballesteros AJ, Menezes R, Allmendinger R (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2019, Springer International Publishing, Cham, pp 150–160
126. Kuwajima H, Tanaka M, Okutomi M (2019) Improving transparency of deep neural inference process. Progress in Artificial Intelligence 8(2):273–285
127. Laidlaw C, Feizi S (2019) Playing it safe: Adversarial robustness with an abstain option. arXiv preprint arXiv:191111253
128. Le H, Voloshin C, Yue Y (2019) Batch policy learning under constraints. In: International Conference on Machine Learning, PMLR, pp 3703–3712
129. Le MT, Diehl F, Brunner T, Knol A (2018) Uncertainty estimation for deep neural object detectors in safety-critical applications. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp 3873–3878
130. Lecuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S (2018) On the connection between differential privacy and adversarial robustness in machine learning. stat 1050:9
131. Lee K, Lee K, Lee H, Shin J (2018) A simple unified framework for detecting out-of-distribution samples and adversarial attacks. 1807.03888
132. Lee K, An GN, Zakharov V, Theodorou EA (2019) Perceptual attention-based predictive control. 1904.11898
133. Lee K, Wang Z, Vlahov B, Brar H, Theodorou EA (2019) Ensemble bayesian decision making with redundant deep perceptual control policies. In: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, pp 831–837
134. Levi D, Gispan L, Giladi N, Fetaya E (2019) Evaluating and calibrating uncertainty prediction in regression tasks. arXiv preprint arXiv:190511659
135. Li J, Liu J, Yang P, Chen L, Huang X, Zhang L (2019) Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification. In: International Static Analysis Symposium, Springer, pp 296–319
136. Li S, Chen Y, Peng Y, Bai L (2018) Learning more robust features with adversarial training. arXiv preprint arXiv:180407757

137. Li S, Shao M, Fu Y (2018) Person re-identification by cross-view multi-level dictionary learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(12):2963–2977, DOI 10.1109/TPAMI.2017.2764893
138. Li Y, Liu Y, Li M, Tian Y, Luo B, Xu Q (2019) D2nn: A fine-grained dual modular redundancy framework for deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference, Association for Computing Machinery, New York, NY, USA, ACSAC '19, p 138–147, DOI 10.1145/3359789.3359831, URL https://doi.org/10.1145/3359789.3359831
139. Liang S, Li Y, Srikant R (2020) Enhancing the reliability of out-of-distribution image detection in neural networks. 1706.02690
140. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:150902971
141. Lin W, Yang Z, Chen X, Zhao Q, Li X, Liu Z, He J (2019) Robustness verification of classification deep neural networks via linear programming. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11418–11427
142. Liu J, Shen Z, Cui P, Zhou L, Kuang K, Li B, Lin Y (2020) Invariant adversarial learning for distributional robustness. arXiv preprint arXiv:200604414
143. Liu L, Saerbeck M, Dauwels J (2019) Affine disentangled gan for interpretable and robust av perception. arXiv preprint arXiv:190705274
144. Liu M, Liu S, Su H, Cao K, Zhu J (2018) Analyzing the noise robustness of deep neural networks. In: 2018 IEEE Conference on Visual Analytics Science and Technology (VAST), IEEE, pp 60–71
145. Lust J, Condurache AP (2020) Gran: An efficient gradient-norm based detector for adversarial and misclassified examples. 2004.09179
146. Lütjens B, Everett M, How JP (2019) Safe reinforcement learning with model uncertainty estimates. In: 2019 International Conference on Robotics and Automation (ICRA), IEEE, pp 8662–8668
147. Lyu Z, Ko CY, Kong Z, Wong N, Lin D, Daniel L (2020) Fastened crown: Tightened neural network robustness certificates. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, pp 5037–5044
148. Ma L, Juefei-Xu F, Zhang F, Sun J, Xue M, Li B, Chen C, Su T, Li L, Liu Y, Zhao J, Wang Y (2018) Deepgauge: Multi-granularity testing criteria for deep learning systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, Association for Computing Machinery, New York, NY, USA, ASE 2018, p 120–131, DOI 10.1145/3238147.3238202, URL https://doi.org/10.1145/3238147.3238202
149. Ma L, Juefei-Xu F, Xue M, Li B, Li L, Liu Y, Zhao J (2019) Deepct: Tomographic combinatorial testing for deep learning systems. In: 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp 614–618, DOI 10.1109/SANER.2019.8668044

150. Machida F (2019) N-version machine learning models for safety critical systems. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), pp 48–51, DOI 10.1109/DSN-W.2019.00017
151. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:170606083
152. Mallozzi P, Pelliccione P, Menghi C (2018) Keeping intelligence under control. In: 2018 IEEE/ACM 1st International Workshop on Software Engineering for Cognitive Services (SE4COG), IEEE, pp 37–40
153. Mani N, Moh M, Moh TS (2019) Towards robust ensemble defense against adversarial examples attack. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp 1–6, DOI 10.1109/GLOBECOM38437.2019.9013408
154. Mani S, Sankaran A, Tamilselvam S, Sethi A (2019) Coverage testing of deep learning models using dataset characterization. 1911.07309
155. Marvi Z, Kiumarsi B (2020) Safe off-policy reinforcement learning using barrier functions. In: 2020 American Control Conference (ACC), IEEE, pp 2176–2181
156. Meinke A, Hein M (2019) Towards neural networks that provably know when they don't know. arXiv preprint arXiv:190912180
157. Meyes R, de Puiseau CW, Posada-Moreno A, Meisen T (2020) Under the hood of neural networks: Characterizing learned representations by functional neuron populations and network ablations. arXiv preprint arXiv:200401254
158. Meyes R, Schneider M, Meisen T (2020) How do you act? an empirical study to understand behavior of deep reinforcement learning agents. arXiv preprint arXiv:200403237
159. Michelmore R, Kwiatkowska M, Gal Y (2018) Evaluating uncertainty quantification in end-to-end autonomous driving control. arXiv preprint arXiv:181106817
160. Mirman M, Gehr T, Vechev M (2018) Differentiable abstract interpretation for provably robust neural networks. In: International Conference on Machine Learning, PMLR, pp 3578–3586
161. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al. (2015) Human-level control through deep reinforcement learning. nature 518(7540):529–533
162. Moravčík M, Schmid M, Burch N, Lisỳ V, Morrill D, Bard N, Davis T, Waugh K, Johanson M, Bowling M (2017) Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. Science 356(6337):508–513
163. Müller S, Hospach D, Bringmann O, Gerlach J, Rosenstiel W (2015) Robustness evaluation and improvement for vision-based advanced driver assistance systems. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp 2659–2664, DOI 10.1109/ITSC.2015.

427
164. Naseer M, Minhas MF, Khalid F, Hanif MA, 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
165. Nguyen HH, Matschek J, Zieger T, Savchenko A, Noroozi N, Findeisen R (2020) Towards nominal stability certification of deep learning-based controllers. In: 2020 American Control Conference (ACC), IEEE, pp 3886–3891
166. Nowak T, Nowicki MR, Ćwian K, Skrzypczyński P (2019) How to improve object detection in a driver assistance system applying explainable deep learning. In: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, pp 226–231
167. O'Brien M, Goble W, Hager G, Bukowski J (2020) Dependable neural networks for safety critical tasks. In: International Workshop on Engineering Dependable and Secure Machine Learning Systems, Springer, pp 126–140
168. Pan R (2019) Static deep neural network analysis for robustness. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Association for Computing Machinery, New York, NY, USA, ESEC/FSE 2019, p 1238–1240, DOI 10.1145/3338906.3342502, URL https://doi.org/10.1145/3338906.3342502
169. Park C, Kim JM, Ha SH, Lee J (2018) Sampling-based bayesian inference with gradient uncertainty. arXiv preprint arXiv:181203285
170. Pauli P, Koch A, Berberich J, Kohler P, Allgower F (2020) Training robust neural networks using lipschitz bounds. arXiv preprint arXiv:200502929
171. Pedreschi D, Giannotti F, Guidotti R, Monreale A, Pappalardo L, Ruggieri S, Turini F (2018) Open the black box data-driven explanation of black box decision systems. arXiv preprint arXiv:180609936
172. Pedroza G, Adedjouma M (2019) Safe-by-Design Development Method for Artificial Intelligent Based Systems. In: SEKE 2019 : The 31st International Conference on Software Engineering and Knowledge Engineering, Lisbon, Portugal, pp 391–397, DOI 10.18293/SEKE2019-094, URL https://hal-cea.archives-ouvertes.fr/cea-02273349
173. Pei K, Cao Y, Yang J, Jana S (2017) Deepxplore. Proceedings of the 26th Symposium on Operating Systems Principles DOI 10.1145/3132747.3132785, URL http://dx.doi.org/10.1145/3132747.3132785
174. Pei K, Cao Y, Yang J, Jana S (2017) Towards practical verification of machine learning: The case of computer vision systems. 1712.01785
175. Peng W, Ye ZS, Chen N (2019) Bayesian deep-learning-based health prognostics toward prognostics uncertainty. IEEE Transactions on Industrial Electronics 67(3):2283–2293
176. Postels J, Ferroni F, Coskun H, Navab N, Tombari F (2019) Sampling-free epistemic uncertainty estimation using approximated variance prop-

agation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 2931–2940
177. Rahimi M, Guo JL, Kokaly S, Chechik M (2019) Toward requirements specification for machine-learned components. In: 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW), pp 241–244, DOI 10.1109/REW.2019.00049
178. Rakin AS, He Z, Fan D (2018) Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. arXiv preprint arXiv:181109310
179. Ramanagopal MS, Anderson C, Vasudevan R, Johnson-Roberson M (2018) Failing to learn: Autonomously identifying perception failures for self-driving cars. IEEE Robotics and Automation Letters 3(4):3860–3867, DOI 10.1109/lra.2018.2857402, URL http://dx.doi.org/10.1109/LRA.2018.2857402
180. Reeb D, Doerr A, Gerwinn S, Rakitsch B (2018) Learning gaussian processes by minimizing pac-bayesian generalization bounds. arXiv preprint arXiv:181012263
181. Remeli V, Morapitiye S, Rövid A, Szalay Z (2019) Towards verifiable specifications for neural networks in autonomous driving. In: 2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo), IEEE, pp 000175–000180
182. Ren H, Chandrasekar SK, Murugesan A (2019) Using quantifier elimination to enhance the safety assurance of deep neural networks. In: 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), IEEE, pp 1–8
183. Ren J, Liu PJ, Fertig E, Snoek J, Poplin R, DePristo MA, Dillon JV, Lakshminarayanan B (2019) Likelihood ratios for out-of-distribution detection. 1906.02845
184. Ren K, Zheng T, Qin Z, Liu X (2020) Adversarial attacks and defenses in deep learning. Engineering 6(3):346–360
185. Revay M, Wang R, Manchester IR (2020) A convex parameterization of robust recurrent neural networks. IEEE Control Systems Letters 5(4):1363–1368
186. Ribeiro FDS, Calivá F, Swainson M, Gudmundsson K, Leontidis G, Kollias S (2019) Deep bayesian self-training. Neural Computing and Applications pp 1–17
187. Ribeiro MT, Singh S, Guestrin C (2016) " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
188. Richards SM, Berkenkamp F, Krause A (2018) The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In: Conference on Robot Learning, PMLR, pp 466–476

189. Roh Y, Heo G, Whang SE (2021) A survey on data collection for machine learning: A big data - ai integration perspective. IEEE Transactions on Knowledge and Data Engineering 33(4):1328–1347, DOI 10.1109/TKDE.2019.2946162
190. Ruan W, Wu M, Sun Y, Huang X, Kroening D, Kwiatkowska M (2019) Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. IJCAI
191. Rubies-Royo V, Calandra R, Stipanovic DM, Tomlin C (2019) Fast neural network verification via shadow prices. arXiv preprint arXiv:190207247
192. Rudolph A, Voget S, Mottok J (2018) A consistent safety case argumentation for artificial intelligence in safety related automotive systems. In: ERTS 2018, Toulouse, France, 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018), URL `https://hal.archives-ouvertes.fr/hal-02156048`
193. Rusak E, Schott L, Zimmermann R, Bitterwolf J, Bringmann O, Bethge M, Brendel W (2020) Increasing the robustness of dnns against image corruptions by playing the game of noise. arXiv preprint arXiv:200106057 3
194. Salay R, Czarnecki K (2018) Using machine learning safely in automotive software: An assessment and adaption of software process requirements in iso 26262. `1808.01614`
195. Salay R, Queiroz R, Czarnecki K (2018) An analysis of iso 26262: Machine learning and safety in automotive software. In: WCX World Congress Experience, SAE International, DOI https://doi.org/10.4271/2018-01-1075, URL `https://doi.org/10.4271/2018-01-1075`
196. Salay R, Angus M, Czarnecki K (2019) A safety analysis method for perceptual components in automated driving. In: 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), IEEE, pp 24–34
197. Sallami MM, Khedher MI, Trabelsi A, Kerboua-Benlarbi S, Bettebghor D (2019) Safety and robustness of deep neural networks object recognition under generic attacks. In: International Conference on Neural Information Processing, Springer, pp 274–286
198. Scheel O, Schwarz L, Navab N, Tombari F (2020) Explicit domain adaptation with loosely coupled samples. `2004.11995`
199. Segú M, Loquercio A, Scaramuzza D (2019) A general framework for uncertainty estimation in deep learning. arXiv preprint arXiv:190706890
200. Sehwag V, Bhagoji AN, Song L, Sitawarin C, Cullina D, Chiang M, Mittal P (2019) Analyzing the robustness of open-world machine learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, Association for Computing Machinery, New York, NY, USA, AISec'19, p 105–116, DOI 10.1145/3338501.3357372, URL `https://doi.org/10.1145/3338501.3357372`
201. Sehwag V, Wang S, Mittal P, Jana S (2020) On pruning adversarially robust neural networks. ArXiv abs/2002.10509

202. Sekhon J, Fleming C (2019) Towards improved testing for deep learning. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), pp 85–88, DOI 10.1109/ICSE-NIER.2019.00030
203. Sena LH, Bessa IV, Gadelha MR, Cordeiro LC, Mota E (2019) Incremental bounded model checking of artificial neural networks in cuda. In: 2019 IX Brazilian Symposium on Computing Systems Engineering (SBESC), IEEE, pp 1–8
204. Sheikholeslami F, Jain S, Giannakis GB (2020) Minimum uncertainty based detection of adversaries in deep neural networks. In: 2020 Information Theory and Applications Workshop (ITA), IEEE, pp 1–16
205. Shwartz-Ziv R, Tishby N (2017) Opening the black box of deep neural networks via information. arXiv preprint arXiv:170300810
206. Singh G, Gehr T, Püschel M, Vechev M (2018) Boosting robustness certification of neural networks. In: International Conference on Learning Representations
207. Sinha A, Namkoong H, Volpi R, Duchi J (2017) Certifying some distributional robustness with principled adversarial training. arXiv preprint arXiv:171010571
208. Smith MT, Grosse K, Backes M, Alvarez MA (2019) Adversarial vulnerability bounds for gaussian process classification. arXiv preprint arXiv:190908864
209. Sohn J, Kang S, Yoo S (2019) Search based repair of deep neural networks. `1912.12463`
210. Somenzi F, Trivedi A (2019) Reinforcement learning and formal requirements. In: International Workshop on Numerical Software Verification, Springer, pp 26–41
211. SOTIF (2019) ISO/PAS 21448: Road vehicles – Safety of the intended functionality. International Organization of Standardization
212. Steinhardt J, Koh PW, Liang P (2017) Certified defenses for data poisoning attacks. `1706.03691`
213. Sumanta Dey BG Pallab Dasgupta (2020) Safety augmentation in decision trees
214. Summers C, Dinneen MJ (2019) Improved adversarial robustness via logit regularization methods. arXiv preprint arXiv:190603749
215. Sun Y, Huang X, Kroening D, Sharp J, Hill M, Ashmore R (2019) Deepconcolic: Testing and debugging deep neural networks. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), pp 111–114, DOI 10.1109/ICSE-Companion.2019.00051
216. Sun Y, Huang X, Kroening D, Sharp J, Hill M, Ashmore R (2019) Structural test coverage criteria for deep neural networks. ACM Trans Embed Comput Syst 18(5s), DOI 10.1145/3358233, URL `https://doi.org/10.1145/3358233`
217. Suri A, Kumaraguru P (2018) Hardening deep neural networks via adversarial model cascades

218. Syriani E, Luhunu L, Sahraoui H (2018) Systematic mapping study of template-based code generation. Computer Languages, Systems & Structures 52:43 – 62, DOI https://doi.org/10.1016/j.cl.2017.11.003, URL http://www.sciencedirect.com/science/article/pii/S1477842417301239
219. Taha A, Chen YT, Misu T, Shrivastava A, Davis L (2019) Unsupervised data uncertainty learning in visual retrieval systems. arXiv preprint arXiv:190202586
220. Tang YC, Zhang J, Salakhutdinov R (2019) Worst cases policy gradients. arXiv preprint arXiv:191103618
221. 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, Association for Computing Machinery, New York, NY, USA, ICSE '18, p 303–314, DOI 10.1145/3180155.3180220, URL https://doi.org/10.1145/3180155.3180220
222. Tian Y, Zhong Z, Ordonez V, Kaiser G, Ray B (2020) Testing dnn image classifiers for confusion & bias errors. 1905.07831
223. Törnblom J, Nadjm-Tehrani S (2020) Formal verification of input-output mappings of tree ensembles. Science of Computer Programming 194:102450
224. Toubeh M, Tokekar P (2019) Risk-aware planning by confidence estimation using deep learning-based perception. arXiv preprint arXiv:191000101
225. Tran HD, Musau P, Lopez DM, Yang X, Nguyen LV, Xiang W, Johnson TT (2019) Parallelizable reachability analysis algorithms for feed-forward neural networks. In: 2019 IEEE/ACM 7th International Conference on Formal Methods in Software Engineering (FormaliSE), IEEE, pp 51–60
226. Tran HD, Yang X, Lopez DM, Musau P, Nguyen LV, Xiang W, Bak S, Johnson TT (2020) Nnv: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In: International Conference on Computer Aided Verification, Springer, pp 3–17
227. Tuncali CE, Fainekos G, Ito H, Kapinski J (2018) Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp 1555–1562, DOI 10.1109/IVS.2018.8500421
228. Turchetta M, Berkenkamp F, Krause A (2016) Safe exploration in finite markov decision processes with gaussian processes. arXiv preprint arXiv:160604753
229. Udeshi S, Jiang X, Chattopadhyay S (2020) Callisto: Entropy-based test generation and data quality assessment for machine learning systems. In: 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST), pp 448–453, DOI 10.1109/ICST46399.2020.00060
230. Uesato J, Kumar A, Szepesvari C, Erez T, Ruderman A, Anderson K, Heess N, Kohli P, et al. (2018) Rigorous agent evaluation: An

adversarial approach to uncover catastrophic failures. arXiv preprint arXiv:181201647
231. Usmanova I, Krause A, Kamgarpour M (2019) Safe convex learning under uncertain constraints. In: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, pp 2106–2114
232. Varghese S, Bayzidi Y, Bar A, Kapoor N, Lahiri S, Schneider JD, Schmidt NM, Schlicht P, Huger F, Fingscheidt T (2020) Unsupervised temporal consistency metric for video segmentation in highly-automated driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp 336–337
233. Wabersich KP, Zeilinger M (2020) Bayesian model predictive control: Efficient model exploration and regret bounds using posterior sampling. In: Learning for Dynamics and Control, PMLR, pp 455–464
234. Wabersich KP, Zeilinger MN (2020) Performance and safety of bayesian model predictive control: Scalable model-based rl with guarantees. arXiv preprint arXiv:200603483
235. Wabersich KP, Hewing L, Carron A, Zeilinger MN (2021) Probabilistic model predictive safety certification for learning-based control. IEEE Transactions on Automatic Control
236. Wagner J, Kohler JM, Gindele T, Hetzel L, Wiedemer JT, Behnke S (2019) Interpretable and fine-grained visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9097–9107
237. Wang J, Cao K, Fang C, Chen J (2019) Fdfuzz: Applying feature detection to fuzz deep learning systems. International Journal of Performability Engineering 15(10):2675
238. Wang J, Gou L, Zhang W, Yang H, Shen HW (2019) Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation. IEEE transactions on visualization and computer graphics 25(6):2168–2180
239. Wang S, Pei K, Whitehouse J, Yang J, Jana S (2018) Efficient formal safety analysis of neural networks. arXiv preprint arXiv:180908098
240. Wang S, Pei K, Whitehouse J, Yang J, Jana S (2018) Formal security analysis of neural networks using symbolic intervals. In: 27th {USENIX} Security Symposium ({USENIX} Security 18), pp 1599–1614
241. Wang TE, Gu Y, Mehta D, Zhao X, Bernal EA (2018) Towards robust deep neural networks. arXiv preprint arXiv:181011726
242. Wang W, Wang A, Tamar A, Chen X, Abbeel P (2018) Safer classification by synthesis. 1711.08534
243. Wang Y, Jha S, Chaudhuri K (2019) Analyzing the robustness of nearest neighbors to adversarial examples. 1706.03922
244. Wang YS, Weng TW, Daniel L (2019) Verification of neural network control policy under persistent adversarial perturbation. arXiv preprint arXiv:190806353
245. Watkins CJ, Dayan P (1992) Q-learning. Machine learning 8(3-4):279–292

246. Wen J, Li S, Lin Z, Hu Y, Huang C (2012) Systematic literature review of machine learning based software development effort estimation models. Information and Software Technology 54(1):41 – 59, DOI https://doi.org/10.1016/j.infsof.2011.09.002, URL http://www.sciencedirect.com/science/article/pii/S0950584911001832
247. Wen M, Topcu U (2020) Constrained cross-entropy method for safe reinforcement learning. IEEE Transactions on Automatic Control
248. Weyuker EJ (1982) On Testing Non-Testable Programs. The Computer Journal 25(4):465–470, DOI 10.1093/comjnl/25.4.465, URL https://doi.org/10.1093/comjnl/25.4.465, https://academic.oup.com/comjnl/article-pdf/25/4/465/1045637/25-4-465.pdf
249. Wicker M, Huang X, Kwiatkowska M (2018) Feature-guided black-box safety testing of deep neural networks. 1710.07859
250. Wolschke C, Kuhn T, Rombach D, Liggesmeyer P (2017) Observation based creation of minimal test suites for autonomous vehicles. In: 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), pp 294–301, DOI 10.1109/ISSREW.2017.46
251. Wu M, Wicker M, Ruan W, Huang X, Kwiatkowska M (2020) A game-based approximate verification of deep neural networks with provable guarantees. Theoretical Computer Science 807:298–329
252. Xiang W, Lopez DM, Musau P, Johnson TT (2019) Reachable set estimation and verification for neural network models of nonlinear dynamic systems. In: Safe, autonomous and intelligent vehicles, Springer, pp 123–144
253. Xie X, Ma L, Juefei-Xu F, Xue M, Chen H, Liu Y, Zhao J, Li B, Yin J, See S (2019) Deephunter: A coverage-guided fuzz testing framework for deep neural networks. In: Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, Association for Computing Machinery, New York, NY, USA, ISSTA 2019, p 146–157, DOI 10.1145/3293882.3330579, URL https://doi.org/10.1145/3293882.3330579
254. Xu H, Chen Z, Wu W, Jin Z, Kuo S, Lyu M (2019) Nv-dnn: Towards fault-tolerant dnn systems with n-version programming. In: 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), pp 44–47, DOI 10.1109/DSN-W.2019.00016
255. Yaghoubi S, Fainekos G (2019) Gray-box adversarial testing for control systems with machine learning components. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, Association for Computing Machinery, New York, NY, USA, HSCC '19, p 179–184, DOI 10.1145/3302504.3311814, URL https://doi.org/10.1145/3302504.3311814
256. Yan M, Wang L, Fei A (2020) Artdl: Adaptive random testing for deep learning systems. IEEE Access 8:3055–3064, DOI 10.1109/ACCESS.2019.2962695

257. Yan Y, Pei Q (2019) A robust deep-neural-network-based compressed model for mobile device assisted by edge server. IEEE Access 7:179104–179117, DOI 10.1109/ACCESS.2019.2958406
258. Yang Y, Vamvoudakis KG, Modares H (2020) Safe reinforcement learning for dynamical games. International Journal of Robust and Nonlinear Control 30(9):3706–3726
259. Ye S, Tan SH, Xu K, Wang Y, Bao C, Ma K (2019) Brain-inspired reverse adversarial examples. 1905.12171
260. Youn WK, Hong SB, Oh KR, Ahn OS (2015) Software certification of safety-critical avionic systems: Do-178c and its impacts. IEEE Aerospace and Electronic Systems Magazine 30(4):4–13, DOI 10.1109/MAES.2014.140109
261. Zhan W, Li J, Hu Y, Tomizuka M (2017) Safe and feasible motion generation for autonomous driving via constrained policy net. In: IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, IEEE, pp 4588–4593
262. Zhang J, Cheung B, Finn C, Levine S, Jayaraman D (2020) Cautious adaptation for reinforcement learning in safety-critical settings. In: International Conference on Machine Learning, PMLR, pp 11055–11065
263. Zhang JM, Harman M, Ma L, Liu Y (2019) Machine learning testing: Survey, landscapes and horizons. 1906.10742
264. Zhang M, Li H, Kuang X, Pang L, Wu Z (2019) Neuron selecting: Defending against adversarial examples in deep neural networks. In: International Conference on Information and Communications Security, Springer, pp 613–629
265. Zhang P, Dai Q, Ji S (2019) Condition-guided adversarial generative testing for deep learning systems. In: 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), pp 71–72, DOI 10.1109/AITest.2019.000-5
266. Zhao C, Yang J, Liang J, Li C (2016) Discover learning behavior patterns to predict certification. In: 2016 11th International Conference on Computer Science & Education (ICCSE), IEEE, pp 69–73