A Safety View on Generalization for Machine Learning

Alexandru Paul Condurache[1,2]
[1] *Driver Experience, Robert Bosch GmbH*
[2] *Institute for Signal Processing, University of Luebeck*
Germany
alexandrupaul.condurache@de.bosch.com

***Abstract*—As the practical footprint of machine learning (ML) constantly enlarges to include even more new application areas, the topic of safety becomes of major concern. Traditional approaches to safety leverage causality. However, due to the correlation-based nature of the currently dominating ML methods, a new take on safety is needed. In this context, we need to answer in a convincing manner the same key question of finding out the root causes of a failure. Generalization is the ability to correctly decide on previously unseen data. Optimizing the generalization ability, which (since always) lies at the heart of ML, clearly implies dealing with generalization failures and is therefore inherently related to safety. In this contribution, we argue that generalization is a key factor to be considered in building a safety argumentation for ML. The focus on generlization shows clarly that two pillars of ML safety are successful design and the avilability of the right data. ML design and in particular safety-aware ML design needs prior knowledge. We discuss modalities to bring prior knowledge to bear in this setup. Getting the right data in particular for those problem setups that enjoy a large degree of variability is problematic in practice. We argue that this can be achieved by devising ways to constantly sample the data distribution of that particular problem space.**

***Index Terms*—Machine Learning, Safety, Generalization.**

I. Introduction

Artificial Intelligence (AI) represents the study of how to create machines, which can perform tasks that when performed by humans require intelligence, for the purpose of supporting or even replacing the humans at such tasks [21]. Intelligence is the ability to infer information from a potentially changing environment, store it as knowledge and use it to generate optimal behavior. By AI we mean here weak AI, i.e., a system that acts like it would think, as opposed to a strong AI that would actually think for itself. AI thus lays at the intersection of several fields, including knowledge representation, deduction, reasoning, learning, with the latter being related to the ability to adapt to changes. The environment represents the problem setup where AI is applied.

AI is implemented in practice by a set of mathematical objects. The parameters of these mathematical objects can be set as in the case of expert systems or learned as in the case of learning systems. To do so, one needs a set $\mathcal{M} = \{m_1, m_2, \ldots\}$ of samples from the data distribution governing the problem setup. Expert systems leverage to a large extent causal chains related the prior knowledge and understanding of the problem setup by the human designer. The learning systems, while benefiting as well to a certain degree from the abilities of the human designer, are engineered to leverage correlations in the available data beyond the direct understanding of the designer and are thus more likely to provide solutions tailored to the data. Expert systems typically require for the development less data samples than the learning systems. Assuming the available data represents a proper sample of the data distribution, learning systems – generally gathered under the term Machine Learning (ML) – offer AI solutions whose performance nowadays usually surpasses that of expert systems on a majority of practical applications. They show improved generalization in the sense that they are better able to correctly handle samples of the data distribution not present in $\mathcal{M}$. This is the main reason for their current pervasiveness.

In general safety is concerned with failure. As a first step we need to understand what the consequences of failure are. This implies taking at least three factors into consideration: the severity of failure, the set of possible reactions to failure and the probability of failure. Then we need to understand what the prerequisites of failure are, more precisely, we need to understand why does a failure of some severity happens with some probability. In traditional safety approaches this last step relies heavily on causality.

What AI is concerned, to a certain degree, a traditional safety approach may be used for expert systems, but for ML, due to its correlation-based nature, we need novel ways to argue for safety [28], [23]. ML failure is mainly about weak generalization [8], hence the strong relationship between the two. Generalization has been in the focus of ML research since always and lots of efforts have been invested in developing of a learning theory leading to performant ML solutions [29]. It is interesting to notice that current learning theories show that prior expert knowledge also plays a significant role in the design of ML solutions [30], [10], [25]. From a safety perspective such a hybrid approach has the appeal of also introducing a certain degree of causality.

The constantly enlarging practical footprint of (supervised) ML lead to a focus on ML safety which is expressed in a large number of publications in various venues from workshops over conferences to journals. Even though a majority of publications approach ML topics from a safety perspective

[16], [27], [3], [23], [28] there are also safety-related topics that animate the ML research community like for example "Adversarial Samples" [15]. When coming towards ML from safety, often the approach is structured in a traditional manner along concepts that are already an integral part of international standards like ISO26262 [12]. As the understanding of ML safety evolved this is at least partially reflected in new standardization approaches like ISO/PAS 21448 [13] or ISO/PAS 8800 [14], however in general, one follows the same structure built along the general development cycle of any item, starting at requirements engineering and going over development to verification and validation [23]. The approach includes besides design-time measures [1], [9], [2] also runtime measures to improve safety, such as the monitoring of the output of an ML solution [18], [19] and makes provisions for redundancy by diversity. Within this paradigm, ML safety lies at the intersection of large set of measures each tailored to some ML aspect that has some meaning with respect to safety.

In this contribution we argue that ML safety needs to be looked upon through the prism of generalization. This allows to approach ML safety in a principled, structured, and unitary manner providing the foundation for cross-pollination between the two academic fields. Ultimately, a majority if not all safety related aspects from above are related to the generalization ability of ML. The learning theory describes how to generate an ML solution such as to optimize the generalization performance and in doing so it addresses relevant safety concerns. Even though by its correlation-based nature the behavior of a fully developed ML-solution in a particular context remains incomprehensible when trying to rely exclusively on causality, the theory of learning provides us with ways to broadly understand and control generalization from a stochastic perspective while at the same time specifying the role prior expert knowledge plays in this context. For this purpose, the learning theory addresses both sampling from the data distribution and how to use the generated sample to design a ML solution. The design of ML algorithms has enjoyed significant attention lately and we have arguably reached the point where given a proper data sample and by leveraging prior knowledge in an application-dependent way, we can come up with an optimal solution. The focus in practical ML should therefore now lay on how and what type of prior knowledge can best be used in an application-dependent way and especially on how to get a proper data sample. In particular when the information content of the problem space (i.e., the logical framework of the problem that is solved by the ML solution) is large, obtaining a proper data sample implies being able to continuously sample the data distribution during the entire development phase of the ML solution. ML design becomes thus a loop where we are ultimately able to approximate the data distribution to the required accuracy given a constantly improving training sample. Thus, from a practical perspective one of the pillars of generalization and thus a major safety argument is the ability to develop ML solutions using continuous sampling.

II. Theoretical Considerations

Next, we introduce the foundations that allow us to establish or at least underline the link between ML safety and generalization. In Section II-A we discuss the key concerns that need to be addressed by various safety measures for ML. In Section II-B we give a short overview on the theory of generalization.

A. ML Safety Concerns

The measures that need to be implemented within a safety argumentation for ML [23] are ultimately related to a set of concerns [28] that impact the performance of ML solutions. These concerns are:

Inappropriate sample. In this case, the development sample which is often divided into a training, a test and a validation set, does not cover the problem space such that we have no chance of estimating the data distribution from it. The reasons for this may vary, including e.g., a misunderstanding of the problem space.

Distribution shift. Should the problem space evolve and change during the lifetime of the ML solution, then at a certain point in time, the development sample will become inappropriate, nolonger being representative for the new problem space.

Missing (long) tail. If the problem space includes very rare events, properly sampling the corresponding long tail of the data distribution is challenging. Ignoring this issue leads again to an inappropriate sample of the problem space.

Brittleness. Sometimes changes in the input that should be irrelevant for the output, lead to the ML solution returning a bad decision.

Quality of labels. Label errors but also insufficient label information afflict the foundation upon which the ML solution is developed. Still, a limited amount of label noise can arguably be instrumental in the development of a performant ML solution.

Wrong metrics. Optimising performance with inappropriate metrics leads to a poor solution for the given ML task.

Separation of train and test. A core assumption to be met during the development of a ML solution is that the train and test data are independent identically distributed (i.i.d.). This assumption is hurt if the test data is identically distributed but not independent of the training data.

Unreliable confidence. It would be desirable that an ML solution should fail only when its internal confidence in the respective decision is low. However, often when a ML solution fails it does so with a high internal confidence.

Incomprehensible behavior. By its correlation nature an ML solution is difficult to understand in a causal manner.

B. Generalization

In the ML setup we observe data generated by some experiment without access to the internal mechanics of the experiment. When developing an ML solution, we try to infer on the internal mechanics using only a (limited) sample from the distribution $\mathcal{D}$ that models the experiment. This underlines

the correlation-based nature of ML, as opposed to a causality-based approach that would address the internal mechanics directly.

Generalization represents the ability to correctly classify previously unseen data and learning theories address the factors that need to be controlled in a learning machine such as to achieve good generalization [8]. It is assumed that an i.i.d. training sample $\mathcal{S} = \{(x_1, l_1), (x_2, l_2), \ldots, (x_m, l_m)\}$ of input-output pairs is available for choosing the best hypothesis on the true functional relationship between the input and the output of the ML problem. The best hypothesis is the one with the smallest generalization error.

The generalization error is itself a random variable that depends on how good can the true data distribution be inferred from $\mathcal{S}$ that is randomly drawn from $\mathcal{D}$. We impose that the probability that $\mathcal{S}$ gives rise to a hypothesis h_S with a generalization error $e_{\mathcal{D}}(h_S)$ over the entire $\mathcal{D}$ larger than ϵ is smaller than δ

$$P_S(e_{\mathcal{D}}(h_S) > \epsilon) < \delta \tag{1}$$

and look for the factors that need to be considered for selecting h_S such that its generalization error is bounded

$$e_{\mathcal{D}}(h_S) < \epsilon(m, H, \delta) \tag{2}$$

The bound $\epsilon(m, H, \delta)$ depends on the factors δ, discussed above H, which is related to the function set from which h_S is selected and m, which is the cardinality of $\mathcal{S}$ and tells us thus that h is probably approximately correct (PAC).

To proceed further we establish next a bound on $P_S(e_{\mathcal{D}}(h_S) > \epsilon)$. We start by observing that the probability that a certain h with a generalization error $e_{\mathcal{D}}(h) > \epsilon$ is consistent and thus decides correctly for all m components of $\mathcal{S}$ is

$$P_S(e_{\mathcal{D}}(h) > \epsilon) \leq (1 - \epsilon)^m \tag{3}$$

We assume that we work with a finite set H of consistent hypotheses that each perfectly handle the training set and all these hypotheses have the same generalization error as h from above. The probability that at least one of them – be it h_S – is consistent with $\mathcal{S}$ is bounded by the sum of individual probabilities according to Boole's inequality. Considering also that $(1 - \epsilon)^m \leq e^{-\epsilon m}$, we obtain

$$P_S(e_{\mathcal{D}}(h_S) > \epsilon) \leq |H| e^{-\epsilon m} \tag{4}$$

where $|H|$ is the cardinality of H.

According to inequality (1), we would like $P_S(\cdot)$ to be less than δ, making use of the bound in inequality (4), we obtain that

$$|H| e^{-\epsilon m} < \delta \tag{5}$$

which leads to

$$\epsilon > \frac{1}{m}\left(ln|H| + ln\frac{1}{\delta}\right) \tag{6}$$

If the probability that the training set gives rise to a hypothesis h_S with a large error, as indicated in inequality (6) is smaller than δ, then with a probability of $1-\delta$ the generalization error is upper bounded as

$$\epsilon(m, H, \delta) \leq \frac{1}{m}\left(ln\mathcal{C} + ln\frac{1}{\delta}\right) \tag{7}$$

This bound relates the size m of the training sample $\mathcal{S}$, the complexity $\mathcal{C} = |H|$ of the function class to which the hypothesis belongs and the probability δ of $\mathcal{S}$ being a proper sample to the generalization error ϵ. It provides thus the foundation for searching for an optimal h, as the one that exhibits the best generalization performance, if $\mathcal{S}$ is i.i.d.

The PAC bound in (7) assumes the hypothesis makes no error on the training set (i.e., the empirical error is zero) and measures the complexity of the hypothesis space by its cardinality $|H|$. This can be further developed to allow for an empirical error k larger than zero. To also cover hypotheses spaces of infinite cardinality, a new measure of complexity is needed in the form of the Vapnik-Chervonenkis (VC) dimension d. The bound becomes then:

$$\epsilon(m, H, \delta) \leq \frac{2k}{m} + \frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4}{\delta}\right) \tag{8}$$

The VC dimension may also be infinite. To handle such cases, the effective VC dimension is used. The effective VC dimension depends on the separability in the data distribution as is it captured in the training sample $\mathcal{S}$ [8]. The effective VC dimension can be considered as depending on the training algorithm as well [10]. In this case, we typically make use of prior knowledge to steer the training process, i.e., the selection of h_S such that this is successful and the required generalization performance for the particular task is achieved. Furthermore, prior knowledge may also be used to improve separability.

The generalization bounds corresponding to these developments look differently than those introduced until now, nevertheless as discussed next, they exhibt the same dependence on a set of key factors for generalization performance.

1) Factors controlling the generalization.: Looking more carefully at inequations (7) and (8) we observe that they both rely on a set of factors for controlling the generalization. These factors are: the empirical error, the (effective) complexity of the hypothesis space, the cardinality of the training sample and the appropriateness of the training sample, under the i.i.d. assumption.

The generalization performance can be understood by analyzing how do these factors influence the respective bounds. At the same time, these factors are related to the ML-safety concerns. For example bad labels coming from an inadequate development environment lead to a bad estimate of the empirical error. Another example is that a training sample of low cardinality is inappropriate, as it will likely not cover the entire variability in the problem space. Yet another example is that using a hypothesis space of unsuited complexity leads to brittleness as it is less likely to correctly model the (complex) similarity/dissimilarity relationships need for a successful decision.

III. A Generalization-Based Safety Argumentation

Analyzing the concerns in Section II-A, we observe that these can be grouped into several categories by their core reasons and related generalization-controlling factors, which are detailed in Section II-B. This is shown in Table I.

What the incomprehensible behavior is concerned, we can do little about it, given the correlation-based nature of ML, however, the other concerns can be addressed over their reasons. This represents the foundation for building a generalization-based safety argumentation.

In general, the addressable concerns lead to poor estimates of the generalization thus afflicting the performance of the ML solution. This underlines the link between ML safety and generalization, as in this case at least, safety does mean performance. Considering the construction of a good development environment a task that may be accomplished with a fair amount of engineering skills, data and design should be in the focus of ML safety. The topic of ML safety and design is discussed in more detail in Section III-A, while the topic of ML safety and data is addressed in Section III-B.

A. Prior knowledge in design and inference

The theory of generalization also shows how prior knowledge can be used to improve design [25], [30], e.g., by controlling the effective capacity. Such an approach is particularly appealing for a safety argumentation as it also introduces a certain amount of causality in the whole setup. Leveraging prior knowledge leads also to interesting modalities of supporting a safety argument by verifying at inference time if an ML method does abide by the prior knowledge cues specific to the target task when reaching a decision. There are several approaches that may be followed in this case, like for example verifying if the decision violates physical laws or checking if the decision obeys known invariance properties and so on.

There are many different ways to leverage prior knowledge in the design and inference of an ML solution [7], [25]. While the largest amount of effort has been spent in using prior knowledge in the design phase, using prior knowledge at inference time has attracted a fair amount of attention lately, in particular in relation to the topic of 'Adversarial Examples'. A unified approach to using prior knowledge in both design and inference may be defined by targeting separability. As shown before, separability is related to effective capacity and thus to generalization. We will discuss next how to approach separability over invariance and give examples of how to do so in design and inference.

Intuitively, in order to separate two pattern from each other, we need to concentrate on what tells the patterns apart and ignore what they have in common. This intuition lays at the foundation of the definition of separability. The input data enjoys the property of separability when the variance between classes is large, while the variance within each class is small [26], either in the raw inputs or in some feature space that may be computed (ideally without loss of information) from these raw inputs. Therefore, we can improve separability by targeting invariance in the representation of each class.

Invariance [22], [4] is related to the prior knowledge stemming from our intuition on separability, as discussed above. Thus we can introduce prior knowledge on some ML task by means of enforcing invariance to specific cues. For example we know a-priori that object recognition from images should be invariant to distortions due to the optical path but also to geometrical transformations of the object such as rotation, translation, etc.

Invariance can be used at design time such as to limit the size of hypothesis space [20], [5], [6], but also at inference time, using the fact that if the input changes along directions of invariance, this should not afflict the ML decision [18]. What inference time is concerned [17], the invariance properties may come from prior expert knowledge or from various design-time considerations [19].

B. Continous sampling of the data distribution

Another major reason for concern what generalization and ML safety is concerned is data. The best possible design still may lead to unsatisfactory generalization performance if the data is unsuited. While for problem spaces of limited variability, the training data can be collected in a limited number of iterations, we argue that for problem space of large variability, data needs to be collected continuously over the lifetime of the ML solution and steps need to be taken to be able to constantly update it in this time.

Continuous sampling definitely has an active-learning [24] flavour to it, but while active learning is mostly concerned with selecting existing instances, continuous sampling is concerned with generating instances. By continuously sampling the data distribution, one ensures that there is a chance that the best possible sample is available at some point during the lifetime of the ML solution. This implies at the same time that the ML solution is stuck in a cycle of development, inference and evaluation up to the point where performance at inference time is so good that no additional development step is needed. Eventually the engineering effort for implementing such an approach may become large. An illustration of "Continuous Sampling" is shown in Figure 1

Key to running this cycle is the evaluation step, where we need to establish what samples are still needed. Depending on the application, the evaluation step may be limited to one or distributed over several modules of the "Continous Sampling" cycle. The evaluation step involves the definition of a set of measures for the appropriateness of data given the ML task. Appropriate data sample are in this case those samples where the ML solution shows limited performance viz. poor generalization. These samples can be selected either at inference time, when the ML solution is used or at test time, when the ML solution is developed. Checking the generalization performance of the ML solution at inference time is possible, for example, with the help of dedicated methods such as [18]. Conversely, data mining [11] and dataset design [9], [2] techniques may be used to select relevant

TABLE I
REASONS FOR ML SAFETY CONCERNS AND CORRESPONDING GENERALIZATION-CONTROLLING FACTORS.

Reason	Concern	Factor
Correlation nature of ML	Incomprehensible behavior	–
Poor development env.	Quality of labels	Empirical error
	Separation of train and test	The i.i.d. assumption
Bad data	Inappropriate sample	Cardinality Training sample suitability
	Distribution shift	Training sample suitability
	Missing (long) tail	Cardinality Training sample suitability
Poor design	Brittleness	Complexity of hypothesis sp. Training sample suitability
	Unreliable confidence	Complexity of hypothesis sp. Training sample suitability
	Wrong metrics	Complexity of hypothesis sp. Empirical error

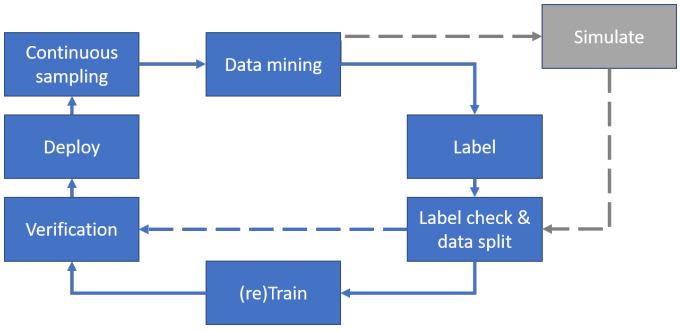

Fig. 1. The "Continuous Sampling" cycle consist of several modules. Mandatory modules are depicted in blue. The "Sampling" module is responsible for gathering data, the "Mining" module is responsible for handling the data (including visualization and knowledge discovery among others), the "Label" module is responsible for generating the ground truth, the "Check & Split" module is responsible for ensuring a high-quality ground truth and a correct split between the train and test datasets, the "Train" module is responsible for establishing the parameters and architecture of the ML solution, the "Verification" module is responsible for verifying that the solution performs satisfactory given the available data, and the "Deploy" module is responsible for the application of the ML solution to the problem space from which the "Sampling" module starts again to gather data. The "Simulate" module, accordingly handles simulation tasks.

samples at a certain iteration of the cycle. Continuous sampling may be enhanced by simulation, for example, in the sense that the moment we become aware that extremely rare samples are needed, we may resort to data augmentation by means of simulation.

IV. DISCUSSION AND CONCLUSIONS

In this contribution we have argued in favor of generalization-based approach to ML safety. We have shown that the core ML safety concerns may be addressed within the learning theory framework that aims at optimal generalization. Starting here, we have identified design (in particular the effective capacity) and data as the main topics to be targeted in a safety argumentation, with data ranking above design. We have also proposed ways to approach these topics by using prior knowledge on invariances to control the effective capacity and verify the decision at inference time and by using continuous sampling to obtain a proper sample of the data distribution.

Once a fully developed ML solution has failed at inference time, it is difficult to understand in a causal manner why it did so, event though to a certain extent, causality can be introduced in this setup over prior knowledge. We can however take measures to ensure that we do not do any obvious errors during the development process, which includes the development setup, the design process and the available data. The development setup should include tools and measures to ensure that the labels are correct, but also that the i.i.d. assumption is respected in train and test data. In the design phase, prior knowledge over invariance properties may be used to control capacity and improve generalization. Producing a proper development sample is arguably the central challenge what ML safety is concerned and continuous sampling should support this.

Even though continuous sampling is theoretically simple, the engineering burden that needs to be overcome to implement it, depending on the target setup is by no means negligible. For example, approaches as continuous sampling are currently being developed in the field of Autonomous Driving (AD), where safety plays a major role. In this case continuous sampling is implemented with the help of a fleet of vehicles that constantly gathers data from the problem space and sends selected samples back to initialize a new development step. In the case of AD, the evaluation step is supported by the fact that currently we have a human in control. The reactions of the human driver may be compared against the reactions of the AI chain implementing AD such as to detect limited-performance sample for which the two would decide differently. Considering that an AD system in general consists of a perceive block that measures the environment and a planning block that computes the part to be followed, also other specific steps can be taken to solve the evaluation issue of continuous sampling in a satisfactory manner. For example, some outputs of the perceive block may be checked against a high-definition map. As soon as the development step is finished the new set of ML solutions are deployed in the fleet and the continuous sampling cycle proceeds.

With proper care, concerning mainly the availability of data from the problem space and the development environment, and eventually with additional measures such as controlling the performance at inference time, ML can definitely be used for safety-critical applications. By optimizing generalization ML has always had safety in focus.

REFERENCES

[1] S. Abrecht, M. Akila, S. S. Gannamaneni, K. Groh, C. Heinzemann, S. Houben, and M. Woehrle, "Revisiting neuron coverage and its application to test generation," in *SAFECOMP 2020 Workshops, Computer Safety, Reliability, and Security*, 2020.

[2] S. Abrecht, L. Gauerhof, C. Gladisch, K. Groh, C. Heinzemann, and M. Woehrle, "Testing deep learning-based visual perception for automated driving," *ACM Transactions on Cyber-Physical Systems*, vol. 5, pp. 1–28, 2021.

[3] S. Burton, L. Gauerhof, and C. Heinzemann, "Making the case for safety of machine learning in highly automated driving," in *SAFECOMP 2017, Computer Safety, Reliability, and Security*, 2017.

[4] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *ICML 2016 International Conference on Machine Learning*, 2016.

[5] B. Coors, A. P. Condurache, and A. Geiger, "Learning transformation invariant representations with weak supervision," in *VISAPP 2018 International Conference on Computer Vision Theory and Applications*, 2018.

[6] ——, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *ECCV 2018 European Conference on Computer Vision*, 2018.

[7] ——, "Nova: Learning to see in novel viewpoints and domains," in *3DV 2019 International Conference on 3D Vision*, 2019.

[8] N. Cristianini and J. Shawe-Taylor, *An introduction to Support Vector Machines*. Cambridge: Cambridge University Press, 2000.

[9] C. Gladisch, C. Heinzemann, M. Herrmann, and M. Woehrle, "Leveraging combinatorial testing for safety-critical computer vision datasets," in *CVPR 2020 Workshops, IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge: MIT Press, 2016.

[11] T. Hastie, R. tibishirani, and J. Friedman, *The elements of statistical learning*, 2nd ed. New York: Springer, 2009.

[12] I. S. O. ISO, "Road vehicles - functional safety," in *ISO 26262*, 2018.

[13] ——, "Road vehicles - safety of the intended functionality," in *ISO/PAS 21448*, 2019.

[14] ——, "Road vehicles - safety and artificial intelligence," in *ISO/PAS 8800*, 2021.

[15] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *ICLR 2017 International Conference on Learning Representations*, 2017.

[16] Z. Kurd and T. Kelly, "Establishing safety criteria for artificial neural networks," *Knowledge-Based Intelligent Information and EngineeringSystems*, vol. 2773, pp. 163–169, 2003.

[17] J. Lust and A. P. Condurache, "A survey on assessing the generalization envelope of deep neural networks: Predictive uncertainty, out-of-distribution and adversarial samples," *CoRR*, vol. abs/2008.09381, 2019.

[18] ——, "Gran: An efficient gradient-norm based detector for adversarial and misclassified examples," in *ESANN2020 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2020.

[19] ——, "Efficient detection of adversarial, out-of-distribution and other misclassified samples," *Neurocomputing*, vol. 470, pp. 335–343, 2021.

[20] M. Rath and A. P. Condurache, "Invariant integration in deep convolutional feature space," in *ESANN2020 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2020.

[21] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2019.

[22] H. Schulz-Mirbach, "On the existence of complete invariant feature spaces in pattern recognition," in *ICPR 1992 International Conference on Pattern Recognition*, 1992.

[23] G. Schwalbe and M. Schels, "A survey on methods for the safety assurance of machine learningbased systems," in *ERTS2020, 10th European Congress on Embedded Real Time Software and Systems*, 2020.

[24] B. Settles, *Active learning literature survey*. TR1648, 2009.

[25] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: from theory to algorithms*. Cambridge: Cambridge University Press, 2014.

[26] C. W. Therrien, *Decision estimation and classification*. New York: John Willey, 1989.

[27] K. R. Varshney, "Engineering safety in machine learning," in *Information Theory and Applications Workshop*, 2016.

[28] O. Willers, S. Sudholt, S. Raafatnia, and S. Abrecht, "Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks," in *SAFECOMP 2020 Workshops, Computer Safety, Reliability, and Security*, 2020.

[29] D. H. Wolpert, *The Mathematics of Generalization*. CRC Press, 1995.

[30] ——, "The lack of a priori distinctions between learning algorithms," *Neural computation*, vol. 8, pp. 1341–1390, 1996.