Training on the Test Set: Mapping the System-Problem Space in AI

José Hernández-Orallo[*1,2], **Wout Schellaert**[*1], **Fernando Martínez-Plumed**[*3,1]

[1]Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València
[2]Leverhulme Centre for the Future of Intelligence, University of Cambridge
[3] European Commission, Joint Research Centre
jorallo@upv.es, wschell@vrain.upv.es, Fernando.MARTINEZ-PLUMED@ec.europa.eu

Abstract

Many present and future problems associated with artificial intelligence are not due to its limitations, but to our poor assessment of its behaviour. Current evaluation practices produce aggregated performance metrics that lack detail and quantified uncertainty about the following question: *how will an AI system, with a particular profile π, behave for a new problem, characterised by a particular situation μ*? Instead of just aggregating test results, we can use machine learning methods to fully capitalise on this evaluation information. In this paper, we introduce the concept of an *assessor model*, $\hat{R}(r|\pi, \mu)$, a conditional probability estimator *trained on test data*. We discuss how these assessors can be built by using information of the full system-problem space and illustrate a broad range of applications that derive from varied inferences and aggregations from $\hat{R}$. Building good assessor models will change the predictive and explanatory power of AI evaluation and will lead to new research directions for building and using them. We propose accompanying every deployed AI system with its own assessor.

Introduction

We will argue that the primal goal of AI evaluation is to predict the performance of an AI system π on a new problem situation μ. Since an evaluation function $R(\pi, \mu)$ can be stochastic, its results r are *measurements* that follow a conditional distribution $R(r|\pi, \mu)$. These measurements are assembled from the evaluation of π (i.e., the test set), but they can also serve as training data for a conditional probability (or density) estimator,

$$\hat{R}(r|\pi, \mu) \;\approx\; \Pr(R(\pi, \mu) = r).$$

Dubbed an "assessor model", this estimator predicts distributions for r given specific values or hypothesised distributions for π and μ. Its estimates are used to assess the system *before* any inference or action takes place.

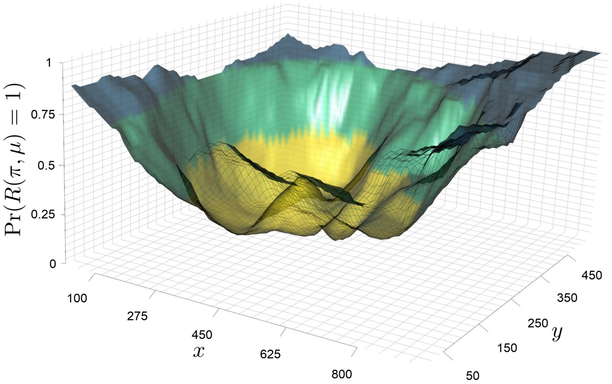

Figure 1: An illustrative example of a delivery robot struggling more in certain destinations than others (e.g., the city centre). By knowing the (x, y) coordinates of the delivery, we can assess in advance whether it is worth deploying the system or not. Other example features making up μ could be weather conditions or package size, while features of the system profile π could be battery level or the exploration criterion. The success or result r is then conditioned on these features and, given a value for all of them, an assessor will estimate a probability of success $\Pr(R(\pi, \mu) = 1)$. We could give distributions or fix values for some features and see the probability map for the rest (x and y in the plot).

A reliable estimation of an AI system's degree of success is an essential element for trust and safety. This estimate serves as an interpretive tool for humans, or as part of a higher-level automated system. For instance, given an autonomous delivery service we can use an assessor $\hat{R}$ to anticipate whether a particular robot π is likely to succeed for a particular delivery μ, as shown in Fig. 1.

This already alleviates key issues in evaluation. First, aggregate performance estimates (e.g., robots are 73.2% successful overall) fail to capture that some situations might just be easier than others. Second, aggregate metrics are not representative when any form of distribution shift or out-of-distribution (OOD) data is present (Quiñonero-Candela et al. 2009; Arjovsky 2020), an increasingly more common problem as systems and tasks become more general (Csordás, Irie, and Schmidhuber 2021; Chen et al. 2020; Hsu et al. 2020; Mohseni et al. 2020; Bevandić et al. 2018; Hendrycks, Mazeika, and Dietterich 2018; Lee et al. 2018). In contrast, the surface in Fig. 1 is given by a model that can interpolate in new areas. Finally, there is much lacking on the explainability front as well; we would like to know *where* a system

[*]These authors contributed equally.

might fail (city centre in the example) leading to possibly also knowing *why* (Bhatt et al. 2021).

The bottom line is that aggregate metrics underuse evaluation data, which instead should be employed to train powerful models of AI system behaviour. Assessor models do just that. Additionally, to fully explore the system-problem space more generally, they should meet the desiderata in Table 1:

Anticipative	It is essential that an assessor is able to *predict performance before a system is dispatched*, even in areas it has never been used.
Standalone	An assessor must *work independently from the original system*, not requiring access to the system or its outputs.
Granular	An assessor must *predict at instance granularity*, and reflect that some situations are easier than others.
Behavioural	An assessor must learn representations of the *emergent behaviour of the system*, without access to system internals (black box approach).
Distributional	An assessor's predictive power will come from populations of (related) systems, but also aggregating *estimates conditioned to distributions*.

Table 1: Desiderata for assessor models.

For all the desiderata above, it is crucial that π and μ have *properties*, i.e., they are tuples $\pi = \langle \theta_1, \theta_2, ..., \theta_i \rangle$ and $\mu = \langle \chi_1, \chi_2, ..., \chi_j \rangle$. For instance in Fig. 1 we had $\pi = \langle \text{battery}, \text{exploration} \rangle$ and $\mu = \langle \text{weather}, \text{package size}, \text{destination x}, \text{destination y} \rangle$. These can be anything that is known about the system or problem; e.g., for π other properties could be deployment conditions, resources available, state, system architecture, or hyperparameters, while for μ they could be the original problem features, goals, or operating conditions (e.g., weather, instance weights).

How to Build an Assessor

While no technique today integrates all the desiderata in Table 1, there are numerous areas that meet some of them, and serve as inspiration and support to generalise from:

- Any model with probabilistic outputs (e.g., class probabilities) is doing a sort of self-assessment with a fixed profile π. Uncertainty estimators can get really sophisticated (Malinin and Gales 2018; Gawlikowski et al. 2021; Gal 2016; Clements et al. 2019; Corbière et al. 2019) while calibration is more lightweight (Song et al. 2019). They are usually neither *anticipative* nor *standalone*.
- AutoML (Hutter, Kotthoff, and Vanschoren 2019) strives to find the features of π that would maximise an aggregate R. Meta-learning focuses on model selection given meta features at the level of datasets (Vanschoren 2018). Both are not applied at the instance level (*granular*) and are rarely used in a *distributional* way.
- Capability-oriented evaluation, machine theory of mind, and machine behaviour explicitly target behaviour, but these areas rarely put the emphasis on being *anticipative* and *distributional* (Rahwan et al. 2019; Rabinowitz et al. 2018; Hernández-Orallo et al. 2016; Hernández-Orallo 2017a,b; Martínez-Plumed et al. 2019; Liao, Zhang, and Chen 2021)
- With quantification and non-additive aggregation, fine-grained predictions can be aggregated by different properties (like a hierarchical roll-up). Quantification has always been applied to the original system (Forman 2008; Bella et al. 2010, 2014; González et al. 2017), but not to a *standalone* assessor model.
- Item Response Theory estimates expected correct response for pairs of subjects and items (Embretson and Reise 2013; Martínez-Plumed et al. 2016; Martínez-Plumed et al. 2019). Because it does not use any property of systems or problems, it is not *anticipative*.

By filling the gaps and connecting the strong points between these areas, we can provide a general tool to unlock the utility of a more widely explored system-problem space. We emphasise this space by considering the data that is used to train an assessor model: a set of empirical measurements $\dot{R} = \{r_1, r_2, ..., r_n\}$, each with its associated system π_i from a population Π and associated instance μ_j from a problem class M. An assessor must thus learn from examples of the form $\langle \pi_i, \mu_j, r_k \rangle$, belonging to an evaluation dataset E. Today, because of the existence of benchmarks, competitions and AutoML scenarios, many problems have been addressed by dozens or hundreds of approaches. Accordingly, evaluation datasets should abound.

The global goal is to minimise the following error:

$$\sum_{\langle \pi, \mu \rangle \in \Pi \times M} \mathcal{D}(\hat{R}(\cdot|\pi, \mu), R(\cdot|\pi, \mu)), \quad (1)$$

where, for a concrete π and μ, $\mathcal{D}$ computes a divergence or loss between an assessor's prediction (which is a distribution) and the true distribution. As we do not know $R(\cdot|\pi, \mu)$, in practice we must minimise the empirical error

$$\sum_{\langle \pi, \mu, r \rangle \in E} \mathcal{S}(\hat{R}(\cdot|\pi, \mu), r) \quad (2)$$

where $\mathcal{S}$ should be a strictly proper scoring rule.

The construction, interpretation and evaluation of an assessor depends on the output space of the evaluation function. Let us consider the simplest case first, where the evaluation function is binary, i.e., $R(\pi, \mu) \in \{0, 1\}$. This happens in problems where there is correct or incorrect output, or a goal that might either be met or not (e.g., classification, theorem proving, etc.). Then, given π and μ, the true distribution $R(\cdot|\pi, \mu)$ would be Bernoulli, defined by a single probability p. For instance, in our delivery example (Fig. 1), a robot π for an instance μ has a 87.2% probability of success. In this case, $\hat{R}(\cdot|\pi, \mu)$ is a conditional probability function that will return a probability $\hat{p}$, which we would like to compare to the true p with a simple $\mathcal{D}$ such as cross-entropy or squared loss. However, as we do not know p, we would use Eq. 2, with S being empirical counterparts: logloss or the Brier score.

The case is more complex when $R(\pi, \mu) \in \mathbb{R}$, since the true distribution $R(\cdot|\pi, \mu)$ can take the form of any continuous distribution. An assessor model will be a conditional density function. We can build a parametric assessor model that assumes that this distribution is Gaussian. Accordingly, this assessor will return two values, a mean m and the variance s^2 for each pair π, μ. Again, instead of calculating the divergence to the true distribution, we use the empirical distribution, as per Eq. 2. Here, for S we could use the log likelihood, or some transformation, such as $(1 + \hat{R}(r|\pi, \mu))^{-1}$ (Hernandez-Orallo 2014).

Assessors are trained from data using some S as loss function, and they may encounter the same issues as any predictive model, including overfitting, calibration and OOD problems. However, we hope the corresponding challenges will be less prominent due to a few reasons: (i) Assessors can take advantage of patterns exclusive to the evaluation space, and the output is less specific. For instance, assessing the success of a system on a blurry image seems easier than choosing one class among twenty. (ii) From an operational perspective it is easier to focus on calibrating these models well, as outputting an accurate distribution is their main utility, not only an afterthought. (iii) They can also specialise on the problem, as separate branches or heads of a neural network can do (Voss et al. 2021; Corbière et al. 2019). (iv) Finally, in many situations we have more features and data than for the original problem, especially when data is available from competitions or AutoML sessions.

In order to illustrate how assessor models can be built, we choose a very simple example with the classification problem segment (Brodley 1990), containing 2310 outdoor images, with 19 attributes each and a class indicating 7 possible types of images. A measurement $R(\pi, \mu) = 1$ if the classification is correct, and 0 otherwise. Through cross-validation, we train a population Π of four neural networks $\{\pi_1, ..., \pi_4\}$ with identical architecture and hyperparameters: one dense hidden layer with size 10 and ReLu activation, softmax activation for the output layer, trained for 15 epochs. We construct a combined evaluation dataset $\dot{R}$ from the cross validation test folds $(4 \times 25\%)$, which we split 75%-25% for training and testing the assessor. The assessor architecture is identical to the systems in the population, but it has the id $\theta \in \{1, ..., 4\}$ of the system as an extra input feature and only a single output representing the estimated probability of success, i.e. $\hat{R}(1|\theta, \mu)$. We also set higher weights for instances with $\theta = 1$, to specialise the assessor for π_1. Fig. 2 reports the aggregated performance (actual and estimated as a quantification task). The Brier Score for $\hat{R}$ and π_1 against the overlapping part of $\dot{R}$ is 0.086 and 0.085 respectively, where a constant baseline always estimating the average accuracy would score 0.124.

Using Assessor Models

Once $\hat{R}$ is built and shown to be an accurate estimator, we can use it to make varied inferences. In Fig. 2 we show an aggregation over instances, but we can just as well use the assessor with a fixed example and average on a distribution of models. For instance, for μ_{82} we get $\mathbb{E}_{\pi \in \Pi} \hat{R}(0|\pi, \mu_{82}) = 0.29$, while μ_{1015} gets 0.89, meaning the former is more difficult for the population Π according to the assessor.

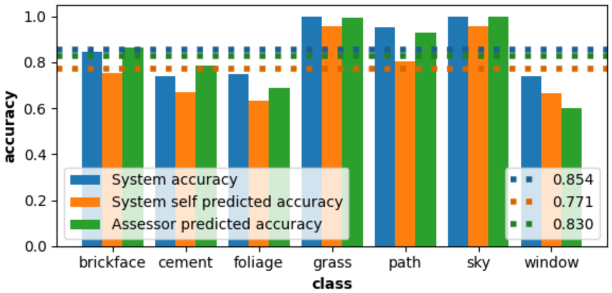

Figure 2: Aggregated estimated results for an assessor model built on the results of four neural networks $\pi_1, ..., \pi_4$ trained on the segment dataset. We show the original accuracy of system π_1, i.e., the average of $R(\pi_1, \mu)$, compared to its self-predicted accuracy, i.e., the average of $\max \pi_1(\mu)$ with π_1 returning a vector with the predicted probabilities of all classes, and the accuracy predicted by the assessor as the average of $\hat{R}(1|\pi_1, \mu)$.

In general, the explanatory and predictive power of an assessor paves the way for a range of applications, some partially covered by more specific solutions, but some completely new. Let us briefly describe some of them:

Predicting instance performance The first use of an assessor is its application to our opening question: *how π is going to behave for μ*. For a binary evaluation function, an assessor is a conditional probability estimator, and trained to predict the probability of success for each instance, $\hat{R}(r = 1|\pi, \mu)$, since the probability of failure is simply $1 - r$. For continuous evaluation functions, an assessor is a conditional density function, and we should ask slightly different questions; for instance, what is the probability that $r \leq r_a$, or what is the probability that $r \in [r_a, r_b]$? If an assessor is parametric, e.g., assuming a normal distribution, then these questions are trivial from the mean and variance returned by an assessor for each pair of π and μ. Equally straightforward is the calculation of confidence intervals.

Predicting populational performance Assessors can range between any extreme of aggregation: measuring a *system's* average performance $\hat{R}(1|\pi, \cdot)$, which is the common practice in AI evaluation, or measuring an *instance's* failure rate $\hat{R}(0|\cdot, \mu)$ (i.e., its difficulty). We can estimate *at any level of aggregation and for any distribution*. For instance, in Fig. 1, we can estimate the distribution of results when $x \sim \mathcal{N}(400, 10)$ and $y \sim U(180, 120)$. If an assessor is well calibrated at the granular level, these aggregations will work well, as we have seen in Fig. 2. However, if an assessor is not well calibrated we may need quantification methods. Selections can also be used for different purposes, to cover dataset shift or specific operating conditions.

Assessors for selecting and combining systems For each new situation, an assessor can be used to make a selection of the system best fit for the job. For instance, given two robots, we can estimate their expected success when $\mu \sim \mathcal{F}$, with

$\mathcal{F}$ being a new distribution. Or in supervised problems, an assessor could be used to give weights to models in an ensemble (Dietterich 2000; Zhou 2019). Unlike many of these approaches, we do not need to run the systems to have their weights. An assessor works standalone, like a conditional computation method where (parts of) systems are also selected in advance (Shazeer et al. 2016; Cheng et al. 2020; Zhang, Chen, and Zhong 2021).

Assessors for anomaly and perturbation detection In other cases, it is interesting to run the system and see how its results $\dot{R}$ compare with what $\hat{R}$ predicted. For instance, a system failing on a batch of instances while $\hat{R}$ predicted differently might be a sign that something strange is happening. This could be an adversarial attack or any other applied distortion which is modifying the original instances (e.g., using different degrees of blur, including adversarial patches or watermarks, adding errors in the direction of the gradient, etc.). Note that adversarial attacks may have little effect on $\hat{R}$, even though the system fails for the given instance. The adversarial attack usually targets the latent features and gradients of the system, but it is not (yet) designed to fool an assessor. Assessors are also useful to understand pockets of instances for which the model works suspiciously well, such as Clever Hans phenomena (Lapuschkin et al. 2019; Hernández-Orallo 2019).

Assessors for explaining failures or fixing them When a system fails, an assessor can be used to identify the features of μ or even of π that likely caused the system to fail. An assessor can be interrogated to determine variations of the properties of π that predict higher results. Also, we can think of small variations of μ that could lead to success (e.g., μ cannot be solved with π but slightly similar μ' or π' would work). Asking questions about these properties as counterfactuals could give very useful information to explain when a system fails, and explore solutions. Actually, XAI methods, such as LIME and many others (Molnar 2020), could be applied to an assessor rather than the original model.

Assessors for AutoML and adaptive sampling A special case of the above is when we search for the best properties of π for a given population of instances. For instance, in AutoML (Hutter, Kotthoff, and Vanschoren 2019), when looking for optimal hyperparameters, the search algorithm can interrogate an assessor as a heuristic rather than running all possibly experiments. Questioning an assessor is much cheaper than training and evaluating a new model. Unlike meta-learning (Vanschoren 2018), AutoML and related approaches, an assessor is granular (level of instance rather than dataset). This implies that it could determine those instances with more variance for different hyperparameters, and concentrate the actual search on them. In other words, assessors can be used for adaptive sampling (Shekhar, Javidi, and Ghavamzadeh 2020) but also for active learning (Settles 2009, 2011; Chakraborty 2020): the most informative instances are those for which an assessor is less certain, i.e., the output distribution has a high variance.

Infer fairness metrics for different distributions Many fairness metrics compare results when conditioned to values of a protected attribute. For instance, *overall accuracy equality* (Verma and Rubin 2018) can be derived from an assessor model by conditioning on the protected attribute and seeing whether the distributions on r change. Using an assessor rather than the data has the advantage that we could try to explain why the model is unfair using XAI techniques on an assessor, e.g., why is this face recognition system worse for this group than for any other group?

Maintenance and revision Any AI system will be subject to changes and monitoring (Sculley et al. 2015; Hernández-Orallo et al. 2016). If these changes can be expressed as properties that change a profile π into π', then an assessor could be retrained to this evolution of the system. Covering this change of systems may lead to much more robust estimates than when using an uncertainty or confidence estimator that needs to be created from scratch for each new revision of the system. It can also highlight the properties responsible for the increase or decrease in performance.

Auditing and certification Intellectual property concerns do not always allow disclosing the inner workings of an AI system. This is problematic regarding transparency and external auditing of the system, and similarly for estimating performance or explaining results during deployment. When it is not possible to share details or run the system (IP issues, compute or hardware costs, etc.), at least an adequate associated assessor model should be shared; one that should be fully open and auditable. Actually, not being able to provide a good assessor for an AI system telling the operating conditions where it succeeds and fails could be a reason to deny the authorisation to release a system (Brundage et al. 2020; Falco et al. 2021). This holds even more so in safety critical domains such as medicine or the automotive.

The Road Ahead

We have argued that the primal goal of AI evaluation is to predict system performance on unseen problems. We have illustrated how assessors can be built and used to explore the system-problem space, but the full potential of learning behavioural models from evaluation data is currently unengaged. We must analyse, sooner rather than later, the tradeoff between the performance of a system and its predictability, similar to how it is done in XAI. As a general formulation, assessors are the ideal framework to do so. They come with operating conditions, uncertainty, and population information naturally included. Safety analysis can be done with an assessor as both reference and as object, and explainability techniques get a new microscope to find *where* and *why* things go wrong. Having every deployed AI system backed by and accounted for with its assessor model will enhance transparency, auditing, and system selection.

In all, the research directions and opportunities are plenty, since the problems they could solve are plenty as well, and these problems already exist today. We therefore hope assessor models will become an active research area in the years to come, as staple companions of every deployed AI system.

Acknowledgements

This work has been partially supported by the EU (FEDER) and Spanish MINECO grant RTI2018-094403-B-C32 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe", Generalitat Valenciana under grant PROMETEO/2019/098, EU's Horizon 2020 research and innovation programme under grant agreement No. 952215 (TAILOR), the Future of Life Institute, FLI, under grant RFP2-152, US DARPA HR00112120007 (RECoG-AI) and the AI-Watch project by DG CONNECT and DG JRC of the European Commission. This work was also funded by the European Union with the "Programa Operativo del Fondo Europeo de Desarrollo Regional (FEDER) de la Comunitat Valenciana 2014-2020" under agreement INNEST/2021/317 (Neurocalçat) and by the Vic. Inv. of the Universitat Politècnica de València under "programa de ayudas a la formación de doctores en colaboración con empresas" (DOCEMPR21).

References

Arjovsky, M. 2020. *Out of distribution generalization in machine learning*. Ph.D. thesis, New York University.

Bella, A.; Ferri, C.; Hernández-Orallo, J.; and Ramirez-Quintana, M. J. 2010. Quantification via probability estimators. In *International Conference on Data Mining*, 737–742. IEEE.

Bella, A.; Ferri, C.; Hernández-Orallo, J.; and Ramírez-Quintana, M. J. 2014. Aggregative quantification for regression. *Data Mining and Knowledge Discovery*, 28(2): 475–518.

Bevandić, P.; Krešo, I.; Oršić, M.; and Šegvić, S. 2018. Discriminative out-of-distribution detection for semantic segmentation. *arXiv preprint arXiv:1808.07703*.

Bhatt, U.; Antorán, J.; Zhang, Y.; Liao, Q. V.; Sattigeri, P.; Fogliato, R.; Melançon, G.; Krishnan, R.; Stanley, J.; Tickoo, O.; Nachman, L.; Chunara, R.; Srikumar, M.; Weller, A.; and Xiang, A. 2021. Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, 401–413. Association for Computing Machinery. ISBN 978-1-4503-8473-5.

Brodley, C. 1990. Image Segmentation Data Set. http://archive.ics.uci.edu/ml/datasets/image+segmentation.

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

Chakraborty, S. 2020. Asking the Right Questions to the Right Users: Active Learning with Imperfect Oracles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3365–3372.

Chen, J.; Li, Y.; Wu, X.; Liang, Y.; and Jha, S. 2020. Informative outlier matters: Robustifying out-of-distribution detection using outlier mining. *arXiv preprint arXiv:2006.15207*.

Cheng, A.-C.; Lin, C. H.; Juan, D.-C.; Wei, W.; and Sun, M. 2020. InstaNAS: Instance-Aware Neural Architecture Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 3577–3584.

Clements, W. R.; Robaglia, B.-M.; Van Delft, B.; Slaoui, R. B.; and Toth, S. 2019. Estimating risk and uncertainty in deep reinforcement learning. *arXiv preprint arXiv:1905.09638*.

Corbière, C.; Thome, N.; Bar-Hen, A.; Cord, M.; and Pérez, P. 2019. Addressing Failure Prediction by Learning Model Confidence. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Csordás, R.; Irie, K.; and Schmidhuber, J. 2021. The Devil is in the Detail: Simple Tricks Improve Systematic Generalization of Transformers. *arXiv:2108.12284 [cs]*. ArXiv: 2108.12284.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, 1–15. Springer.

Embretson, S. E.; and Reise, S. P. 2013. *Item response theory*. Psychology Press.

Falco, G.; Shneiderman, B.; Badger, J.; Carrier, R.; Dahbura, A.; Danks, D.; Eling, M.; Goodloe, A.; Gupta, J.; Hart, C.; et al. 2021. Governing AI safety through independent audits. *Nature Machine Intelligence*, 3(7): 566–571.

Forman, G. 2008. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2): 164–206.

Gal, Y. 2016. Uncertainty in deep learning. Technical report, University of Cambridge.

Gawlikowski, J.; Tassi, C. R. N.; Ali, M.; Lee, J.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; et al. 2021. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*.

González, P.; Castaño, A.; Chawla, N. V.; and Coz, J. J. D. 2017. A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50(5): 1–40.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*.

Hernandez-Orallo, J. 2014. Probabilistic Reframing for Cost-Sensitive Regression. *ACM Transactions on Knowledge Discovery from Data*, 8(4): 1–55.

Hernández-Orallo, J. 2017a. Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement. *Artif. Intell. Rev.*, 48(3): 397–447.

Hernández-Orallo, J. 2017b. *The Measure of All Minds: Evaluating Natural and Artificial Intelligence*. Cambridge University Press.

Hernández-Orallo, J. 2019. Gazing into Clever Hans machines. *Nature Machine Intelligence*, 1.

Hernández-Orallo, J.; Martínez-Plumed, F.; Schmid, U.; Siebers, M.; and Dowe, D. L. 2016. Computer models solving intelligence test problems: Progress and implications. *Artificial Intelligence*, 230: 74–107.

Hernández-Orallo, J.; Martínez-Usó, A.; Prudêncio, R. B.; Kull, M.; Flach, P.; Farhan Ahmed, C.; and Lachiche, N. 2016. Reframing in context: A systematic approach for model reuse in machine learning. *AI Communications*, 29(5): 551–566.

Hsu, Y.-C.; Shen, Y.; Jin, H.; and Kira, Z. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10951–10960.

Hutter, F.; Kotthoff, L.; and Vanschoren, J. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.

Lapuschkin, S.; Wäldchen, S.; Binder, A.; Montavon, G.; Samek, W.; and Müller, K.-R. 2019. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications*, 10(1): 1096.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Liao, Z.; Zhang, P.; and Chen, M. 2021. ML4ML: Automated Invariance Testing for Machine Learning Models. *arXiv:2109.12926 [cs]*.

Malinin, A.; and Gales, M. 2018. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501*.

Martínez-Plumed, F.; Prudêncio, R. B.; Martínez-Usó, A.; and Hernández-Orallo, J. 2016. Making sense of item response theory in machine learning. In *European Conference on Artificial Intelligence (ECAI), Best Paper Award*, 1140–1148. IOS Press.

Martínez-Plumed, F.; Prudêncio, R. B. C.; Usó, A. M.; and Hernández-Orallo, J. 2019. Item response theory in AI: Analysing machine learning classifiers at the instance level. *Artif. Intell.*, 271: 18–42.

Mohseni, S.; Pitale, M.; Yadawa, J.; and Wang, Z. 2020. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5216–5223.

Molnar, C. 2020. *Interpretable machine learning*. Lulu. com.

Quiñonero-Candela, J.; Sugiyama, M.; Lawrence, N. D.; and Schwaighofer, A. 2009. *Dataset shift in machine learning*. Mit Press.

Rabinowitz, N. C.; Perbet, F.; Song, H. F.; Zhang, C.; Eslami, S. M. A.; and Botvinick, M. 2018. Machine Theory of Mind. *arXiv:1802.07740 [cs]*.

Rahwan, I.; Cebrian, M.; Obradovich, N.; Bongard, J.; Bonnefon, J.-F.; Breazeal, C.; Crandall, J. W.; Christakis, N. A.; Couzin, I. D.; Jackson, M. O.; Jennings, N. R.; Kamar, E.; Kloumann, I. M.; Larochelle, H.; Lazer, D.; McElreath, R.; Mislove, A.; Parkes, D. C.; Pentland, A. S.; Roberts, M. E.; Shariff, A.; Tenenbaum, J. B.; and Wellman, M. 2019. Machine Behaviour. *Nature*, 568(7753): 477–486.

Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.-F.; and Dennison, D. 2015. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28*, 2503–2511.

Settles, B. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Settles, B. 2011. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, 1–18. JMLR Workshop and Conference Proceedings.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2016. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *International Conference on Learning Representations*.

Shekhar, S.; Javidi, T.; and Ghavamzadeh, M. 2020. Adaptive sampling for estimating probability distributions. In *International Conference on Machine Learning*, 8687–8696. PMLR.

Song, H.; Diethe, T.; Kull, M.; and Flach, P. 2019. Distribution calibration for regression. In *International Conference on Machine Learning*, 5897–5906. PMLR.

Vanschoren, J. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*.

Verma, S.; and Rubin, J. 2018. Fairness definitions explained. In *2018 IEEE/ACM international workshop on software fairness (fairware)*, 1–7. IEEE.

Voss, C.; Goh, G.; Cammarata, N.; Petrov, M.; Schubert, L.; and Olah, C. 2021. Branch Specialization. *Distill*, 6(4): e00024.008.

Zhang, Y.; Chen, Z.; and Zhong, Z. 2021. Collaboration of Experts: Achieving 80% Top-1 Accuracy on ImageNet with 100M FLOPs. *arXiv:2107.03815 [cs]*. ArXiv: 2107.03815.

Zhou, Z.-H. 2019. *Ensemble methods: foundations and algorithms*. Chapman and Hall.