Measuring Beyond Accuracy

Violet Turri, Rachel Dzombak, Eric Heim, Nathan van Houdnos, Jay Palat, Anusha Sinha

Introduction

Most machine learning (ML) projects focus on "accuracy" for model evaluation. While accuracy is useful for knowing how well a model performs on a test dataset at the time of model development, there are other significant implications in assessing the utility and usability of a machine learning model. Key considerations include robustness, resilience, calibration, confidence, alignment with evolving user requirements, and fit for mission and stakeholder needs as part of integrated system, among others. In this paper, we explore what it means to measure beyond accuracy and define critical considerations for the test and evaluation of machine learning and, more broadly, artificial intelligence (AI) systems. After defining key measurement considerations, the AI engineering community will be better equipped to develop and implement comprehensive and applied methods for the evaluation of models as well as possible metrics for more realistic and real-world model evaluation.

Current Assessment Practices for AI Systems

Modern AI systems, many of which are built using machine learning, are a departure from static software systems that yield deterministic results. In contrast to analytical systems that follow explicit "instructions" given by a programmer and can be reduced and decomposed, AI systems are empirical, opaque, and unpredictable — they behave based on what they "learn" from data or experience (Russel & Norvig, 2021). Because AI systems often model real-world relationships, they must be adaptable to changing inputs and shifting correlations.

The characteristic differences between traditional software and AI lead to a series of open questions around the design and implementation of AI. For example, AI systems can evolve and change behavior over time. How can we ensure that AI systems are still doing what they are supposed to do? How can we certify that they are safe and reliable? Many current AI and ML methods are data intensive; continuous updates to data, while necessary, can impact architectural concerns such as prediction accuracy and latency (Ozkaya, 2020). Technical debt, driven by data dependencies, accumulates rapidly, silently, and at the system level (Sculley et al., 2015). What practical mechanisms exist to evaluate the state of a system when using large and evolving data sets? Further, exhaustive testing is currently not possible for systems that learn and adapt. How do we change approaches to test and evaluation to be more risk-, resilience-, and process- focused rather than exhaustive?

A common practice for engineering AI systems is to select, optimize, and measure one or more metrics throughout the development pipeline. Common metrics, where applicable, include accuracy, precision, recall, ROC curves, confusion matrices, mean squared error (MSE), and/or mean absolute error (MAE) (Handelman et al., 2019). While the optimization of metrics in software development is not unique to AI, AI is exceptionally good at performing optimizations. Although properly defined and comprehensive metrics can yield impressive quantitative results, excessive optimization of inadequate metrics can result in manipulation, gaming, and/or a focus on short-term quantities, in addition to other potential negative consequences (Thomas & Uminsky, 2020). Utilizing incomplete and/or misleading metrics to test and evaluate AI systems is therefore fraught with risk.

Today, questions around the design, development, implementation, and sustained management of AI are examined across a variety of fields including software engineering, human-centered design, computer science, and systems engineering. We believe that to build AI as well as it can be done, a whole-systems approach is needed. As stated by Ackoff and Wardman (2016), “when you take a system apart, it loses all of its essential properties.” In this paper, we outline holistic considerations for testing and evaluation and aim to extend beyond common practice by capturing cross-disciplinary perspectives on AI engineering, acknowledging the volatility of relying heavily on metrics, and addressing the unique challenges of working with numerous evolving, interconnected system components.

Characteristics of AI Systems

AI Engineering is a field of research and practice that combines cross-disciplinary perspectives to create AI systems in accordance with human needs for stakeholder outcomes. When thinking about measurement of systems, it’s important to start with defining the attributes of a system that are desired. Here, we draw on the three pillars of AI Engineering to guide our thinking on test and evaluation strategy:

1. Human-Centered AI: Implementing AI in context requires a deep understanding of the individuals that intend to use and interact with the system. From a human-centered perspective, systems should be evaluated to assess the alignment with humans, their behaviors, and values, as well as the utility of systems to achieve stakeholder-driven outcomes.
2. Scalable AI: Many current AI and ML methods are compute-intensive, expensive, and time-consuming to develop, necessitating consideration of how to scale AI and ML techniques to real-world size and complexity. In the context of scalability, evaluation lenses could include how AI infrastructure, data, and models may be reused across problem domains and deployments and increase performance to support operational needs. Another evaluation lens could include how effectively a system scales up to support more needs at the enterprise level or scale down to enable capabilities in edge contexts.
3. Robust and Secure AI: AI and ML introduce new system failure modes, vulnerabilities, and attack vectors and change over time. Assurance is needed that systems will work as expected when faced with uncertainty or threat. The Robust and Secure AI pillar provides the context for how we develop and test AI systems to ensure resilience across contexts and when encountering new phenomena over time.

With these pillars in mind, we examine core considerations that AI engineers and AI engineering teams must take into account when developing test and evaluation strategies for AI systems. Due to the prevalence of ML systems in use today, we will provide many of the considerations in the context of ML.

Considerations in Assessing AI Systems

1. What are the intentions of your testing?

While many of the questions listed below are not unique to AI systems in the abstract, each becomes unique when attempting to answer them in the context of AI systems. Unlike in traditional software, the answers to each of these questions are typically less definitive due to the non-deterministic nature of AI. An AI system may perform as expected given one input but behave in an unintended manner on a similar input; testing cannot be exhaustive due to the singular nature of any one data point. As a result, there is a limit to the possible coverage of testing in the context of AI systems.

A helpful starting point for AI engineers is to consider what possible sources of uncertainty exist in a system of interest. In image classification, for instance, the system makes class predictions with varying degrees of confidence, but typically only the prediction with the highest confidence is returned to the end-user. Additional information about uncertainty is helpful during the testing process. Some classes may be hard to distinguish between, even for humans, in which case the model’s limitations mirror human performance. Other mix-ups, for instances between classes that look completely dissimilar, could be indicative of more critical model deficiencies.

Brainstorming potential sources of risk can be another informative practice. What are the potential negative and positive outcomes that the system can produce? How can you evaluate for these incomes, both indirectly and directly? T&E allows for a thorough exploration of the model's behavior before it is deployed and is an opportunity to measure and mitigate sources of risk. This practice is important because it upholds the principles of ethical AI, but it also may be a requirement if a governing entity requires regulatory or legal tests.

Overall, as AI engineers start their exploration into assessment of systems, there are many directions they could go in and realistically, they have both limited time and resources. Setting intentions and learning goals around T&E can help teams to prioritize what they are investing resources into at different times and ensure they are getting actionable information from the testing that is done.

2. What logistical challenges do you need to consider?

Developing a well-documented plan for handling the logistical challenges of T&E for AI systems is another task that is necessary to perform early on. Evaluations can be performed at a variety of levels. AI systems require traditional software T&E-style verification that code is performing properly and is free of bugs; this form of testing can be performed by standard software developers. However, aspects of AI systems that require expertise, such as interpreting the meaning of increased uncertainty in deployment or recognizing the emergence of new classes, may require specialized team members to be involved in the evaluation. Outlining all relevant evaluation concerns and assembling a diverse team to tackle T&E tasks spanning different risk levels and content areas is a crucial step in producing a robust T&E strategy.

Due to the cyclical, interconnected nature of ML lifecycle phases, issues discovered through T&E can have wide-reaching affects for ML systems and plans to mitigate these impacts will likely require communication across phases. The development of ML systems has been mapped to various lifecycle frameworks. Andrew Ng, for instance, educates practitioners to follow a four-part ML lifecycle that includes: (1) Scoping the project, (2) Collecting data (3) Training a model, and (4) Deploying in production (DeepLearningAI, 2021). Garcia et al. (2018) contends that the ML lifecycle consists of context, defined as "all the information surrounding the use of data in an organization," plus three phases: (1) Pipeline Development, (2) Training, and (3) Inference. While differences in proposed frameworks exist, what is true across frameworks is that the ML lifecycle consists of numerous phases intertwined via feedback loops.

Although testing is often represented as a distinct phase of the AI system lifecycle, we advocate that T&E is most effective when integrated throughout all phases. When implementing AI systems, teams of humans must engage in continuous oversight and frequently reflect on the questions: What are we doing? Why are we doing it, and for whom? (Barmer et al., 2021). Accounting for T&E considerations across every stage of AI system development and deployment supports the rapid, iterative development of robust, ethical mission capabilities (JAIC, 2020). Selecting metrics that accurately assess the system's ability to fulfill mission goals and using these metrics throughout training, for instance, will guide the system towards meeting stakeholder needs. A common misconception is that testing is overly time-consuming, while the process of fixing errors, especially late in system development, is what actually absorbs time (Kohavi et al., 2009). Developing a culture of frequent testing and a practice of addressing errors as they arise is an effective method for catching issues early on and preventing a build-up of problems. As stated by Thomke (2020), "Culture—not tools and technology— prevents companies from conducting the hundreds, even thousands, of tests they should be doing annually and then applying the results."

In settings where personnel across the pipeline work together closely, building T&E into all stages of development is realizable. However, when different stakeholders are siloed across the pipeline, such as designers, data scientists, software engineers, machine learning researchers, and operations teams, challenges communicating between roles can be a source of ML "mismatch" (Lewis et al., 2021). Facilitating effective communication across roles on AI teams and throughout organizations to ensure that errors and problems found during T&E are addressed properly is a challenge, especially when elements of the ML pipeline may be handled by different organizations or teams. Methods for addressing errors in T&E may include system rollback (especially in high-risk, mission-critical contexts) or revising the training dataset to account for new information about model performance (Dunnmon et al., 2021).

Constraints on communication and/or access to information across development and deployment pipeline stages should also be accounted for. For example, if provenance or other details about the training data are not accessible by the T&E team this can make it difficult to detect bias and uncertainty. Determining and documenting what aspects of the model and pipeline are within the scope of testing, and in turn what issues and topics can realistically be addressed, is an important step in fleshing out a T&E strategy. Limitations on T&E personnel's access to data or training/deployment specifics can be significant obstacles that may require modifications to team structures and/or documentation procedures.

3. What are your biggest sources of risk?

In the context of AI-enabled systems, it's important to frame risk, or the possibility of suffering loss, in the context of the role that AI is performing within the system (Dzombak et al., 2021; Dorofee et al., 1996). For instance, if an AI component stops working or begins to operate poorly, how will this impact the system's overall ability to perform its task? What does poor operation look like and how can it be measured? Enumerating potential threats to the system, the likelihood of each threat occurring, and the impact of each of these threats early on will provide the T&E team with an estimate of risk that can guide the focus of testing (Alberts & Dorofee, 2010).

Traditional methods for estimating the impact of loss (Kambic et al., 2020; Tucker, 2020) can be applied to AI systems, but estimating the likelihood of loss in the context of AI is an open challenge. Nascent methods for determining the quantitative likelihood of loss can be pulled from an emerging body of work related to AI threat modeling and vulnerabilities (Biggio & Roli, 2018; Beieler, 2019; Householder et al., 2020a; Householder et al., 2020b; MITRE, 2020). In the absence of quantitative estimates, qualitative assessments from domain experts can be leveraged to gauge the relative importance of threats.

Since AI systems have the potential to be used for different tasks, understanding the specific use-cases for which the system will be employed can sharpen the objectives of risk-related testing. For example, consider an overhead object detection system that can identify vehicles of interest to military personnel. This system could be used for at least two different tasks with differing parameters: (1) reconnaissance, a mission that is limited in time and scope, or (2) surveillance, a longer-term mission with less time pressure. The rate of false positives is a critical metric for the reconnaissance use-case because spurious hits could overwhelm an analyst using the system in a time-critical context. On the other hand, the rate of false negatives is pressing for the surveillance task because the user has time to review detections and requires thorough coverage of possible incidents. Considering the specific use-case for a system can determine which metrics are most relevant.

4. What is the meaning behind your metrics?

A challenge in interpreting and selecting metrics for T&E is determining their meaning and impact in *context*. Donella Meadows (1998) stated: "Indicators arise from values (we measure what we care about) and they create values (we care about what we measure)". What is the overall value provided by the AI system and what kinds of measurements can be used to assess progress towards providing this intended value? What impact will prioritizing certain metrics have on the system development? Often, teams implement the measurement systems that they have knowledge of, whether or not they provide the needed meaning. In AI systems, garnering meaning from metrics is complicated by factors such as system complexity, risk, and audience.

AI systems have exceptionally powerful optimization capabilities, therefore the optimization of metrics that do not align with intended values can build systems with impressive test results but produce behaviors that are both unintentional and consequential. For example, Facebook's goal is to drive social connection and their corresponding metrics for measuring connection include time spent on the platform, the number posts users interact with, and how many ads users click on. In the process of optimizing these metrics, their algorithm learned to show users posts that upset or anger them. While the engagement metrics may have improved, their progress towards the goal ultimately suffered. This example demonstrates how defining system goals and choosing metrics for T&E that truly support these values is critical.

While intended system value may be clear, goals towards achieving this value, as well as the metrics for measuring progress towards goals, can be competing or misaligned. For instance, a facial recognition system will likely have to make tradeoffs between achieving high efficiency and fairness across protected groups. Tradeoffs may involve other ethical issues such as privacy, transparency, and accountability (Amarasinghe et al., 2021). Identifying and assessing tradeoffs between metrics is a challenge which remains ongoing throughout the ML system lifecycle.

Assurance that metrics are calculated accurately is a prerequisite to deriving meaning from your measurements. Standard metrics such as accuracy or false positive rates are relatively easy to verify, especially in the presence of clear ground-truth labels. Confidence scores, on the other hand, are a more complex and often unverified metric. While confidence scores can be a valuable source of information about model performance during T&E, these estimates are only useful if they have been calibrated to suggest the true correctness likelihood (Guo et al., 2017). Mechanisms for producing front-facing metrics, such as confidence scores, must be validated prior to deployment to ensure that end-users are given precise information when interacting with the system.

When interpreting performance metrics for non-technical audiences specifically, another set of challenges and opportunities arise. Scores such as F1 or AUC ROC can be difficult to interpret without a technical background in AI; how can these metrics be translated into plain English in the context of the problem at hand? Meaning must be derivable from system metrics not only by ML practitioners, but by other key collaborators involved in designing and reviewing the system. Efforts must also be made to avoid information overload by condensing relevant information and presenting results in a clear, simple, and balanced manner that is accessible for non-technical stakeholders (IDF, 2020).

Additionally, it's important to consider what metrics requirements really mean and how they align with project objectives. For instance, if a decision threshold was used, how can this cutoff be justified and was this decision appropriate for the goal? Fan and Lin (2007) discuss how performance metrics can be improved by changing decision thresholds. Modifying thresholds, however, can result in a selection rate that does not make sense for your problem. Perspectives from domain experts, where applicable, can help shape discussions around metric expectations and parameters including threshold.

5. How are you dealing with scale and the level of complexity in your system?

A significant challenge facing AI system developers today is how to create systems that can operate across a variety of domains and use cases. Success is hard enough to achieve when operating AI systems in closely controlled development and laboratory environments, and even more challenging when considering scale and system complexity.

Achieving the development and deployment of robust and secure AI systems requires the creation of new T&E strategies that take scope into account. To thoroughly evaluate system performance, T&E teams must acknowledge that an AI system will not exist in a vacuum; consider how the system operates and what kinds of interactions will take place between the system and sub-systems or contextual systems of interest. What inputs will the system receive and what outputs are expected? What impact will faulty results or predictions have on downstream components? How does the system respond to invalid inputs? This context can inform expectations about system behavior and sources of potential risk and, in turn, guide the selection of appropriate metrics and methods for addressing these requirements.

Furthermore, it's important to consider and routinely reevaluate the setting(s) in which the AI system will be employed and what use in these environments entails. What differences there are between your local test environment and global implementation contexts? In deployment, a system may require different computer resources to meet increased demand or operate on a delayed retraining schedule. Likewise, the system may encounter different real-world relationships between inputs and outputs, unexpected input data, or distinct types of end-users. To prepare for diverse use-cases, training and testing data must address a variety of real-world scenarios.

That being said, comprehensive coverage is likely unattainable. For complex systems, it's impossible to generate a complete list of scenarios in which the system may fail (Doshi-Velez & Kim, 2017). Instead of working towards a "perfect" system, T&E teams can build confidence in an AI system through rigorous testing, evaluation, verification and validation (TEV&V) incorporated throughout the system's lifecycle. While in traditional software operational metrics are the primary concern for system monitoring, in the realm of AI engineering performance metrics are also important (Huyen, 2022). Monitoring a system in deployment contexts and tracking new behavior such as data drift will provide teams with the information, they need to retrain a system to meet emerging needs and to tune performance expectations to reflect new requirements.

6. How are you evaluating implications for humans and unintended consequences?

To ensure that a system is responsible and equitable, it must be vetted during T&E for unintended and/or negative consequences on the humans who will be impacted by the system. Across all phases of AI system development, it's important to keep in mind who will be engaging with the system, directly or indirectly, and what they will potentially gain or lose through their interactions. If the system is designed to support or replace an existing process, tests that reflect existing expectations and best practices in deployment should be developed alongside domain experts. What pain-points for the user within the current system, and to what degree will the new system improve upon or change how such issues manifest? Likewise, what potentially negative tradeoffs exist in the new system and how will they impact the user?

ML models learn the relationships explicitly or implicitly embedded within their (often historical) training dataset; as a result, they have the potential to pick up on undesirable correlations between inputs and outputs. The existence of unintended relationships between features in the training dataset can cause the model to "learn" the wrong thing altogether. Geirhos et al. (2021) describes one such example in which an image classifier identifies cows based on the presence of grass in the image; while the model achieved high performance accuracy, the model failed on examples in which a cow was pictured in a new setting. When working with human-related inputs, such as in facial recognition systems, correlations can include racial or gender bias, among other demographic disparities. Unintentional correlations in training data can have real-world impacts; New Jersey's pretrial risk assessment algorithm, for instance, was trained on data that "reflects racial and ethnic disparities in policing, charging, and judicial decisions" and, as a result, made decisions that "perpetuate racial inequalities" with regards to detainment (Simonite, 2020).

Analyzing training data directly is a powerful method for detecting and preventing biases and other unwanted correlations; the REVISE tool (Wang et al, 2020), for example, presents techniques for identifying and mitigating a variety of biases in visual datasets. The Gender Shades project provides an illustrative example of how comparing model performance across different demographics of subjects can reveal underlying system biases (Buolamwini & Gebru, 2018). The detection and mitigation of bias is a crucial concern for AI systems working with human subjects, but systems designed for other input types can also exhibit biases, such as producing better results for certain languages or geographic inputs than others. Careful analysis of training, evaluation, and testing data in the early stages of model development is a crucial preventative measure towards smart testing and evaluation. Repeated analysis across stages, especially if data is collected in deployment and used to retrain the model, is another important quality check. Explainability techniques can also be used to periodically probe the model and determine which features are the most important factors in determining system outputs. Auditing of a system before its adoption is critical to prevent unwanted consequences.

An important caveat to consider throughout bias mitigation efforts is the risk of accidentally masking unfair behavior through selected metrics. The Propublic Machine Bias study (Angwin et al., 2016) provides a case study of how metrics measured on data "slices" can conceal discriminatory model behavior. In the study, researchers examined a "fair" model for predicting crime recidivism that received similar accuracies across different racial groups. Upon closer examination, however, researchers discovered that Black defendants were twice as likely to be falsely identified as recidivists, while White defendants were twice as likely to be falsely identified as non-recidivists. The decision to use accuracy alone to identify bias resulted in real-world racially discriminatory practices; this study illustrates the importance of conducting in-depth analysis of model behavior instead of settling for surface-level results.

Iterating on your T&E strategy

Documentation of your T&E strategy during the early phases of planning will prove critical as you iterate on the approach. After a first attempt at evaluation, it's important to reconsider what the main tradeoffs are within the evaluation strategy and determine if there are any important system or model attributes that are currently unaccounted for. Throughout the lifecycle of the model or system, different needs may appear as data inputs and/or expected outputs change. Proper monitoring of the system (often considered the final stage of the ML pipeline) is necessary to recognize changes such as data drift or concept drift. Information gleaned about current project needs through monitoring can be used to ensure that the T&E procedure covers relevant risks and concerns.

For instance, imagine an ML speech-to-text model for making song requests as part of a music streaming system. Developers focus on training and testing a model that achieves high accuracy on speech samples from young users, as this is the platform's target demographic. The system performs well for the first six months but sees an increase in middle-aged users in the second half of the year and a corresponding drop in average accuracy. While the initial goal for T&E was to ensure that the system achieved high accuracy for young users, the top priority will now likely shift towards achieving acceptable accuracy across varied ages. Dataset curation for both training and testing must grow to include samples from middle-aged speakers and, in anticipation for future new users, developers should consider including a larger range of speech samples across different demographics (e.g., age, dialect).

Other possible changing project needs could include increased or decreased scale, the emergence of adversaries, or the introduction of a new class. T&E considerations will likely fluctuate in priority depending on these needs. Keeping inventory of which aspects of the system are thoroughly tested as well as topics for future testing will be crucial to developing and maintaining a robust and up-to-date strategy.

Conclusion

In conclusion, testing for accuracy alone is not enough to assess the correctness or quality of a ML model. To engineer robust and secure, scalable, and human-centered AI systems T&E needs to account for potential sources of risk and uncertainty early on and incorporate testing measures that address these concerns across all stages of development and deployment. This approach differs from typical AI T&E approaches that view testing as a distinct stage in a linear pipeline and instead opts for a holistic vision of testing that considers the connections between phases of the model lifecycle. Since comprehensive testing is impossible for AI systems, it's important to determine the intentions behind testing and to make informed tradeoffs. Maintaining documentation of process, iterating on T&E strategies in response to emerging requirements, and developing diverse teams to handle varied testing responsibilities are practices that can improve both the depth and breadth of testing. As AI engineering best practices continue to evolve, the delta between traditional systems and AI systems will be further explored and addressed.

Acknowledgements

The authors would like to acknowledge Carol Smith for her invaluable input enumerating human-centered concerns for consideration six.

References

Ackoff, R., & Wardman, K. (2016, August 16). Systems Thinking: What, Why, When, Where, and How? Retrieved from https://thesystemsthinker.com/systems-thinking-what-why-when-where-and-how/.

Alberts, C., & Dorofee, A. (2010). Risk Management Framework (CMU/SEI-2010-TR-017). Retrieved December 09, 2021, from the Software Engineering Institute, Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9525.

Amarasinghe, K., Casey, P., Driscoll, A., Ghani, R., Jones, C., & Rodolfa, K. (2021). Data Science Project Scoping Guide. Data Science and Public Policy Lab at Carnegie Mellon University. Retrieved from http://www.datasciencepublicpolicy.org/our-work/tools-guides/data-science-project-scoping-guide/.

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016, May 23). Machine Bias. Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Barmer, Hollen, Rachel Dzombak, Matthew Gaston, Vijaykumar Palat, Frank Redner, Carol Smith, and Tanisha Smith. (2021). "Human-Centered AI." SEI White Paper.

Beieler, J. "AI Assurance and AI Security: Definitions and Future Directions," presented at the Adversarial Machine Learning Technical Exchange, Rockville, MD, Sep. 24, 2019, [Online]. Available:https://cra.org/ccc/wpcontent/uploads/sites/2/2020/02/John-Beieler_AISec_AAAS.pdf.

Biggio B., & Roli, F. (2018) "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, Jan. 2018, pp. 2154–2156, doi: 10.1145/3243734.3264418.

Buolamwini, J., & Gebru, T. (2018, January). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency* (pp. 77-91). PMLR.

Dorofee, A., Walker, J., Alberts, C., Higuera, R., Murphy, R., & Williams, R. (1996). Continuous Risk Management Guidebook. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996 http://www.sei.cmu.edu/library/abstracts/books/crmguidebook.cfm.

DeepLearningAI. (2021, March 24). *A Chat with Andrew on MLOps: From Model-centric to Data-centric AI* [Video]. YouTube. https://www.youtube.com/watch?v=06-AZXmwHjo.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608.*

Dunnmon, J., Goodman, B., Kirechu, P., Smith, C., & Van Deusen, A. (2021, November). Responsible AI Guidelines in Practice. Retrieved from https://www.diu.mil/responsible-ai-guidelines

Dzombak, R., Krishnan, R., Smith, C., Tucker, B., & VanHoudnos, N. (2021) Comments on the NIST AI Risk Management Framework RFI. Software Engineering Institute, Carnegie Mellon University.

Fan, R. E., & Lin, C. J. (2007). A study on threshold selection for multi-label classification. *Department of Computer Science, National Taiwan University*, 1-23.

Garcia, R., Sreekanti, V., Yadwadkar, N., Crankshaw, D., Gonzalez, J. E., & Hellerstein, J. M.. Context: The missing piece in the machine learning lifecycle. In KDD CMI Workshop, volume 114, 2018.

Geirhos, R., Jacobsen, J. H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), 665-673.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017, July). On calibration of modern neural networks. In *International Conference on Machine Learning* (pp. 1321-1330). PMLR.

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., … & Asadi, H. (2019). Peering Into the Black Box of Artificial Intelligence: Evaluation Metrics of Machine Learning Methods. *American Journal of Roentgenology, 212*(1), 38-43. doi:10.2214/ajr.18.20224

Householder, A., Spring, J., and VanHoudnos, N. (2020a). On managing vulnerabilities in AI/ML systems. In *Proceedings of New Security Paradigms Workshop* (NSPW ’20). ACM, New York, NY, USA URL https://doi.org/10.1145/3442167.3442177.

Householder, A., Spring, J., VanHoudnos, N., and Wright, O. (2020b). Machine learning classifiers trained via gradient descent are vulnerable to arbitrary misclassification attack. https://kb.cert.org/vuls/id/425163/.

Huyen, C. (2022) *Lecture 10. Data Distribution Shifts and Monitoring* [Lecture Notes] Stanford University. https://docs.google.com/document/d/14uX2m9q7BUn_mgnM3h6if-s-r0MZrvDb-ZHNjgA1Uyo/edit

Interaction Design Foundation. (2020). “Information Overload, Why it Matters and How to Combat It”, Retrieved from https://www.interaction-design.org/literature/article/information-overload-why-it-matters-and-how-to-combat-it

JAIC. (2020, May 27). “JAIC Spotlight: The JAIC’s Test, Evaluation, and Assessment Team Shapes Future AI Initiatives,” Retrieved from https://www.ai.mil/blog_05_27_20-jaic_spotlight_test_evaluation_and_assessment_team.html.

Kambic, Daniel., Moore, Andrew., Tobar, David., & Tucker, Brett. (2020). *Loss Magnitude Estimation in Support of Business Impact Analysis* (CMU/SEI-2020-TR-008). Retrieved December 09, 2021, from the Software Engineering Institute, Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=650828.

Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J. L., & Melamed, T. (2009). Online experimentation at Microsoft. *Data Mining Case Studies*, *11*(2009), 39.

Meadows, D. (1998). Indicators and Information Systems for Sustainable Development. *The Earthscan Reader in Sustainable Cities,* 364-393. doi:10.4324/9781315800462-21

MITRE | ATLAS. (2021). Retrieved December 9, 2021, from https://atlas.mitre.org/

Thomas, R., & Uminsky, D. (2020). The problem with metrics is a fundamental problem for ai. *arXiv preprint arXiv:2002.08512*.

Lewis, G. A., Bellomo, S., & Ozkaya, I. (2021). Characterizing and Detecting Mismatch in Machine-Learning-Enabled Systems. *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. doi:10.1109/wain52551.2021.00028

Ozkaya, I. (2020). "What Is Really Different in Engineering AI-Enabled Systems?," in IEEE Software, vol. 37, no. 4, pp. 3-6, July-Aug. 2020, doi: 10.1109/MS.2020.2993662.

Thomke, S. (2020). Building a culture of experimentation. *Harvard Business Review*, *98*(2), 40-47.

Tucker, B. (2020). *Advancing Risk Management Capability Using the OCTAVE FORTE Process* (). Retrieved December 09, 2021, from the Software Engineering Institute, Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=644636

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Harlow: Pearson.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., ... & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, *28*, 2503-2511.

Simonite, T. (2020, February 19). Algorithms Were Supposed to Fix the Bail System. They Havent. Retrieved from https://www.wired.com/story/algorithms-supposed-fix-bail-system-they-havent/

Stanford Alumni. (2021, November 16). *System Error: Where Big Tech Went Wrong and How We Can Reboot* [Video]. YouTube. https://youtu.be/HFxMpeOXUpk

Wang, A., Narayanan, A., & Russakovsky, O. (2020, August). REVISE: A tool for measuring and mitigating bias in visual datasets. In *European Conference on Computer Vision* (pp. 733-751). Springer, Cham.

DM22-0219