Second Workshop on Metacognitive Prediction of AI Behavior

Combinatorial Testing Applications for Artificial Intelligence Metacognition

Erin Lanus, *Virginia Tech National Security Institute, Arlington, VA, 22203, USA*
Laura J. Freeman, *Virginia Tech National Security Institute, Arlington, VA, 22203, USA*

Abstract—Understanding the limitations of AI-enabled systems in high consequence systems is critical due to the advanced capabilities of AIES, especially those with autonomous functions, and the settings in which they are deployed. Combinatorial testing (CT) offers approaches for test and evaluation (T&E) of AIES to answer questions about the operating envelope of the model, how it might perform outside of the envelope, how to know when it is leaving the envelope, and how to design test sets that distinguish a model's performance on learned domains from its generalization to new domains it might encounter in deployment. Potential exists for CT-based T&E approaches to be designed into the AIES to support metacognition through self-evaluation.

Artificial Intelligence (AI) and especially the subfield of machine learning (ML) are increasingly used in high consequence AI-enabled systems (AIES) such as those within the defense sector. AI may be added to systems to reduce the human footprint in dangerous settings (e.g., making logistic resupply trucks autonomous), as a force multiplier (e.g., machine-assisted detections and tracking for intelligence, surveillance, and reconnaissance), or to enable future low-effort extensibility of a system without explicit reprogramming (e.g., adapting to changing adversary tactics in electronic warfare). Understanding the limitations of AI in high consequence systems is critical due to the advanced capabilities of these systems, especially those with autonomous functions, and the settings in which they are deployed. Recognizing that the increased capability afforded by AI does not come without risk, the Department of Defense (DOD) defines five ethical principles for AIES: responsible, equitable, traceable, reliable, and governable [1]. The governable principle requires that AIES detect unintended behavior with the ability to disengage to avoid unintended consequences. The reliable principle requires that AIES have "explicit, well-defined uses" and be subject to testing across the AIES lifecycle.

A primary reason for using AIES instead of software with programmed logic is adaptability to conditions beyond those encountered pre-deployment. Metacognition, the ability of a system to perform self-analysis or "think about what it thinks," can include a component of metacognitive regulation, controlling learning through understanding its "knowledge gaps" [2] that occur during deployment into new or changing environments.

Metacognition has been proposed for AI safety with a discussion of four safety solution strategies [3]. Inherently safe design involves incorporating an understanding of how the system might fail into the requirements and designing mitigations into the system. This is potentially intractable with AIES operating in the real world. Safety reserves involves validating training datasets to ensure they are representative and complete and rigorously testing the trained model. In addition to a potentially infeasible test budget, this strategy requires that the expected operating environment be fully known pre-deployment; if the environment changes, the training data may no longer be adequate and the test coverage may no longer be complete. Safe fail involves a mechanism that detects when the AIES' error rate is too high and recovers the system to a safe state. Procedural safeguards employs external evaluation methods during operation. Test and Evaluation (T&E) has approaches for all except inherently safe design. One difference between metacognition and T&E is whether the model of the system's performance is internal or external. For example, the fail safe mechanism can be a human operator or other external system that monitors and overrides the AI or, in metacognition, a part of the AI system that performs self-diagnosis and disengages. Metacognition could be used for out of distribution detection (OOD) to flag sit-

uations that were not seen in training. A metacognition model could have a detailed understanding of system performance to identify where and when failures occur. The T&E methods and metrics used for OOD and measuring system performance for fault analysis may be internalized to become metacognitive capabilities.

A generic metacognitive architecture, the metacognitive loop (MCL) addresses robustness issues in AIES with three phases – note, assess, guide [4]. The note phase sets expectations and uses sensory or contextual indicators to detect anomalies, changes to the conditions under which the system has known good performance, as a violation of expectations. The assess phase determines the severity of the anomaly and performs fault location, seeking probable causes for the anomaly. The guide phase makes recommendations to recover and monitors for whether the response results in returning to good system performance. T&E methods and metrics may provide functionality that support all three phases. Known system performance measured across the operating envelope can provide information to detect anomalous performance, and characterizations of the input space can provide information to determine when the system is in unfamiliar territory. Correlations of either the input space or intermediary outputs with system performance can support impact assessment, and T&E methods support root cause analysis and fault location. Last, iterative evaluation can support monitoring system recovery success.

T&E offers approaches to reason about the performance and behavior of AI systems. *What are the dimensions of the operating envelope? Can we know prior to leaving it? Can we transfer a model to this domain? How can we distinguish a model's performance on learned domains from its generalization to new domains? What are the top information gain data points for improving the model for this deployment?*

TEST AND EVALUATION

T&E is the systematic process of characterizing the properties of a system and determining suitability for a purpose. Verification testing compares a system against the requirements specification ("did we build the system right?") while validation testing compares the system against the user's needs ("did we build the right system?"). AIES are often characterized in terms of correctness using performance metrics such as accuracy and F-measure. Properties such as fairness, explainability, interpretability, interoperability, scalability, throughput, safety, and security may also be assessed through T&E. The DOD uses operational test to determine if systems are operationally effective, suitable and survivable for intended use, and lethal in the case of weapon systems.

T&E approaches for complex software systems are typically applicable to testing AI. The VTP framework extends the "Systems V" model to include testing throughout deployment as AI models are updated to respond to changes in mission objective, degradation of the system, or adversarial attack, as well as outlining a process to guide development of a comprehensive test plan [5]. The phase of the overall system lifecycle is used to identify the phase of test; a subsystem considered a unit (e.g., an AI model) for the larger AIES itself is decomposable with its own phases of test. The fields of study that contribute to the system under test (SUT) (e.g., computer science and statistics for the learning algorithm, signal processing for electronic warfare, and psychology for the human-machine team) combined with the hierarchy of test inform the goals of the test (e.g., measuring model accuracy versus measuring trust or cognitive load). The hierarchy of test informs test plan efficiency which, along with the goals of the test, informs the test methods.

The NIST AI Risk Management Framework (AI RMF) describes how AI risks differ from traditional software [6]. They can be organized into risks arising from data quality, nature of AI models, and AI use leading to specific risks of test insufficiency.

Data quality. The data used to train and test AI may not be an accurate representation of the operational environment. The data may be biased, complex, of large volume, stale, or detached from the original context. There is additional privacy risk due to enhanced data aggregation by AI capabilities. Last, supervised learning techniques and most test techniques rely on "ground truth" from data labeled by a human or other system with verified correctness; human labeled data is often difficult to get or non-existent due to labeling cost, and the need for an AI system to make inferences often implies the non-existence of another system or algorithm that can correctly label the same data.

Nature of AI models. AI models may be more complex with greater scale than traditional software due to the complexity of functions they represent. Pre-trained models may be reused for new purposes or transferred to different operating environments for which they are not suitable, and risks or other issues may not be communicated or propagated forward. Last, emergent properties of AI models make predicting failure modes challenging, and opacity of models as learned weights rather than encoded logic makes the fault location T&E process more difficult.

AI use. The use of AI systems leads to post-deployment challenges with maintenance, and risk may arise from data, concept, or model drift. AI or

the surrounding environment may change more rapidly than more static software or hardware systems, and the meaning of concepts that AI is intended to model may change over time. Maintenance for AIES is more frequent than for other software due to drift.

Risk to testing. These risks lead to test difficulties, increasing the risk of test insufficiency. The data and model are both difficult to white-box analyze. Additionally, there is a lack of the use of best practices for data, model, and evaluation documentation. Last, the lack of development controls common in software leads to difficulty determining when or what to test. Variation in outputs of software is due to code changes, whereas variations in outputs for AI can result from changes to the training data, test data, or code.

MITRE's Systems Engineering Processes to Test AI Right (SEPTAR) describes differences in T&E between AI and traditional software [7]. Two differences that lead to challenges establishing test adequacy for AI are non-determinism and implicit requirements.

Non-determinism. Traditional software is deterministic, so test cases that previously passed should continue to pass if the code is unchanged, and there is a singular "correct" answer for an input. Many subfields of AI utilize statistical learning and produce systems that are deterministic at time of inference but give different outputs for similar inputs or inputs that differ in unmeasured ways. Still, some AI systems, especially generative AI, are probabilistic and select from a set of possible outputs with defined probabilities. AIES often approximate a function and give answers that are "close enough," requiring a threshold for what constitutes a correct answer in order to use "all-or-nothing" performance measures that count (e.g., the number of true positives). There may be no single correct answer (e.g., language translation). Thus, testing AI may require a larger test corpus due to needing large amounts of test data to estimate system correctness.

Implicit requirements. Sufficient test sets for both software and AI need to cover all conditions that could impact system performance; however, unlike software that has requirements, AI requirements are often implicitly derived from the training data, and conditions that impact the system may not be known up front. Even which conditions are present in an input sample may be difficult to characterize. AIES are intended to operate in an "open universe" so there is a resistance to defining boundaries, and a large corpus of training data is often used in hopes that the "law of large numbers" mean all conditions will be covered. Both of these add difficulty to characterizing the data and limit the use of approaches for test adequacy measurement from software (e.g., input space partitioning).

T&E requires that the system use cases or requirements be bounded in order to have explainable performance in a well-defined operating environment, yet a primary reason to employ AI is its ability to generalize outside of the training data or learn to adapt to new settings. Many trustworthy AI characteristics are related to the input space due to AI's data dependence. AI should be robust to small changes in the input space with security against adversarial inputs. Unintended data bias that would prevent the AI from learning the true function and generalize outside of the training data should be mitigated. Thus, characterization of the input space, while difficult, is important for T&E.

COMBINATORIAL TESTING

Combinatorial testing (CT) is a black-box approach with demonstrated effectiveness in detecting software failures due to component interactions. The SUT is represented by an input parameter model (IPM) specifying the k system components and their values, v_i for the i^{th} component. Test suites such as covering arrays are generated where a test is an assignment of values to components. CT requires discrete component values. CT is pseudo-exhaustive; the strength t determines the size of combination considered. Combinatorial coverage metrics provide measures of test adequacy; total t-way coverage describes the proportion of t-way component interactions appearing in a test suite [8]. Covering arrays have the property that, for any set of t components, all interactions of those components appear in some row in the array. Covering arrays are efficient; the number of rows necessary in the test suite grows logarithmically in k for fixed t and v.

AI models map the input space to the output space by learning relationships between combinations of features and labels; functions where a single feature is predictive for an output generally do not require AI. Combinatorial coverage $CC_t(\mathcal{D})$ has been adapted to measure gaps in coverage of the input space by dataset $\mathcal{D}$ [9]. Here, the IPM is constructed over the k features of the input space and their values representing the universe of the AIES, and each row of a combinatorial array is the feature values present in a given sample. For deep learning where feature engineering is not performed, metadata may act as a surrogate for features in the sample and so combinatorial coverage is computed over rows of metadata. Consider the simple example binary dataset in Table 1. For $t = 2$, there are $\binom{3}{2}$ feature combinations. Of the $\binom{3}{2}2^2 = 12$ 2-way interactions, 10 are present, so $CC_2 = 0.83$. Missing interactions are {(Hair, Yes), (Live Birth, No)} and {(Live Birth, No), (Eco, Ocean)}. Set difference combinatorial coverage $SDCC_t(\mathcal{A} \setminus \mathcal{B})$

TABLE 1. Animal classification dataset with binary features.

Hair	Live Birth	Eco	Class
No	Yes	Ocean	Orca
Yes	Yes	Woods	Wolf
Yes	Yes	Ocean	Otter
No	No	Woods	Owl

measures the proportion of t-way interactions present in dataset $\mathcal{A}$ that are absent from dataset $\mathcal{B}$ [9].

Measuring and identifying gaps in the combinatorial coverage of the input space has a number of applications for AI T&E. Computing $CC_t(\mathcal{D})$ over the training dataset $\mathcal{D}$ gives an estimate of the operating envelope by enumerating the interactions seen in training. Transfer learning is the process of deploying AI developed in a source domain $\mathcal{S}$ to a target domain $\mathcal{T}$ with no or minimal modification. Computing $SDCC_t(\mathcal{T} \setminus \mathcal{S})$ may indicate when a model can be transferred. Training a classification model on imagery of planes in Southern California and transferring to Northern California imagery exhibited a drop in performance with zero shot transfer, while there was no drop transferring the opposite direction. The Northern imagery dataset covered 10% more 2-way interactions present in the Southern imagery than the South covered of the Northern interactions [9]. Additionally, this provides a mechanism to intelligently select or collect samples to close the gap between source and target domains and improve the model for the target deployment. Retraining on the source dataset augmented with a small number of images from the target domain selected by set differencing achieved higher accuracy than augmenting with the same number selected randomly [10]. Coverage can inform test set design beyond random sampling. Given black-box knowledge of the training dataset, test sets are designed to cover the interactions of the intended operating envelope. With white-box knowledge, SDCC can be used to design representative or challenging datasets. Performance on a modified MNIST dataset was higher on test samples with all feature interactions covered by the training set and lower on those with uncovered interactions [11].

Understanding the dimensions of the operating envelope is necessary for computing the boundaries and determining when the AI is leaving it. Not all (metadata) features or their interactions impact a model, and including non-important feature interactions or excluding important ones may lead to misleading coverage results. Systematic inclusion/exclusion is an approach based on CT and Design of Experiments to identify impactful features [12]. The approach constructs a universal test set that covers all interactions. For each feature interaction, it constructs a training dataset that excludes that interaction, trains a model from scratch, and evaluates the model on the universal test set partitioned into a set of samples that contains the excluded interaction and a set that does not. All other aspects of the training process (e.g., training dataset size, hyperparameters, and architecture) are kept constant. The performance deltas between covered and uncovered test partitions are analyzed, and feature interactions for which performance differs statistically are considered to be important. For a satellite imagery dataset with labeled aircraft, average pan resolution, biome, and hour of day impacted both precision and recall, season impacted only recall, and off nadir max impacted neither. In addition to efficiency gained by limiting combinations considered to strength t, identifying impactful features and features interactions through systematic pre-deployment test may reduce dimensionality for coverage computations in post-deployment monitoring.

FUTURE WORK

CT has been demonstrated as an external evaluation mechanism for AI, primarily during pre-deployment testing with proposed applications for post-deployment monitoring. Workshop feedback on the ideas in this short paper will be used to identify opportunities to design CT-based T&E approaches into the AI system for self-evaluation to answer the same questions through metacognitive capabilities. *Is this situation unfamiliar? Has my performance degraded? What data do I need to improve for this mission?*

Additionally, future work for CT-based T&E of AIES would provide functionality within the MCL framework. Specifically, early combinatorial coverage metrics were binary, measuring presence or absence of interactions, but statistical learning employed by many AIES is impacted by frequency of appearance. Preliminary work has been completed for measuring frequency coverage and differences in combinatorial frequency, but more is needed to support the note phase. CT utilizes covering arrays to determine whether faults up to a specified strength t exist in a software system, but these may not distinguish between interactions to perform fault location. Adaptive testing may be used to generate new tests to distinguish interactions always appearing together and in only faulty tests, or locating arrays may be used when the test suite must be designed up front. CT-based approaches to fault location in AIES have not been applied, to the best of our knowledge. Additionally, CT may have application in performing severity assessment of data or model drift in support of the assess phase.

ACKNOWLEDGMENTS

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense, Director, Operational Test and Evaluation (DOT&E) through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-24-D-0023.

REFERENCES

1. K. Hicks, "Memorandum for senior pentagon leadership commanders of the combatant commands defense agency and dod field activity directors: Implementing responsible artificial intelligence in the department of defense," May 2021.
2. J. A. Crowder and N. Shelli Friess MA, "Metacognition and metamemory concepts for AI systems," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer, Engineering and Applied Computing (WorldComp), 2011, pp. 1–6.
3. B. Johnson, "Metacognition for artificial intelligence system safety – an approach to safe and desired behavior," *Safety Science*, vol. 151, p. 105743, 2022.
4. M. Schmill, T. Oates, M. L. Anderson, D. Josyula, D. Perlis, S. Wilson, and S. Fults, "The role of metacognition in robust AI systems," in *Workshop on Metareasoning at the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008.
5. E. Lanus, I. Hernandez, A. Dachowicz, L. J. Freeman, M. Grande, A. Lang, J. H. Panchal, A. Patrick, and S. Welch, "Test and evaluation framework for multi-agent systems of autonomous intelligent agents," in *2021 16th International Conference of System of Systems Engineering (SoSE)*, 2021, pp. 203–209.
6. E. Tabassi, "Artificial intelligence risk management framework (AI RMF 1.0)," 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936225
7. J. Lockett, F. Reeder, I. Chen, C. Pomales, C. Balhana, D. Moore, and R. Ferguson, "Systems engineering processes to test AI right (SEPTAR)," in *2024 IEEE AUTOTESTCON*. IEEE, 2024, pp. 1–10.
8. D. R. Kuhn, I. Dominguez Mendoza, R. N. Kacker, and Y. Lei, "Combinatorial coverage measurement concepts and applications," in *2013 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2013, pp. 352–361.
9. E. Lanus, L. J. Freeman, D. R. Kuhn, and R. N. Kacker, "Combinatorial testing metrics for machine learning," in *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2021, pp. 81–84.
10. S. F. Ahamed, P. Aggarwal, S. Shetty, E. Lanus, and L. J. Freeman, "ATTL: An automated targeted transfer learning with deep neural networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–7.
11. T. Cody, E. Lanus, D. D. Doyle, and L. J. Freeman, "Systematic training and testing for machine learning using combinatorial interaction testing," in *2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2022, pp. 102–109.
12. E. Lanus, B. Lee, L. Pol, D. Sobien, J. Kauffman, and L. J. Freeman, "Coverage for identifying critical metadata in machine learning operating envelopes," in *2024 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. IEEE, 2024, pp. 217–226.

Erin Lanus is a research assistant professor at the Virginia Tech National Security Institute and Affiliate Faculty Computer Science, Virginia Tech. Her research interests include AI assurance leveraging combinatorial interaction testing, as well as metric and algorithm development for AI T&E with a focus on dataset quality. Lanus received a Ph.D. in computer science from Arizona State University. Contact her at lanus@vt.edu.

Laura J. Freeman is a research professor of statistics, deputy director of the Virginia Tech National Security Institute, and assistant dean for Research for the College of Science, Virginia Tech. Her research interests include experimental methods for conducting research that brings together cyberphysical systems, data science, and AI to address critical challenges in national security. Freeman received a Ph.D. in statistics from Virginia Tech. Contact her at laura.freeman@vt.edu.