AIRC PERSPECTIVES

SEPTEMBER 2022

Best Practices for Addressing New Challenges in Testing and Evaluating Artificial Intelligence Enabled Systems

Dr. Laura Freeman, Dr. Justin Kauffman, Mr. Daniel Sobien, Dr. Tyler Cody, and Dr. Erin Lanus

VIRGINIA TECH NATIONAL SECURITY INSTITUTE

VIRGINIA TECH

INTRODUCTION

The integration of artificial intelligence (AI) and statistical machine learning (ML) into complex systems exposes a variety of challenges in traditional test and evaluation (T&E) practices. As more decisions at varying levels are handled by AI-enabled systems (AIES), we need T&E processes that provide a basis for ensuring system effectiveness, suitability, and survivability. This involves methods for assessing the component ML models and AI algorithms, including the ability to show how they result in repeatable and explainable decisions, as well as an understanding of any failure modes and failure mitigation techniques. Moreover, there is a need for AI assurance to certify that AI algorithms operate as intended and are free of vulnerabilities arising either from faulty design or from adversarially inserted data or algorithm code. T&E needs new processes for characterizing the training data sufficiency for ML models, algorithm and model performance, system performance, and operational capabilities. Freeman (2020) outlined challenges facing current T&E methods for complex software-enabled systems, key challenges exacerbated by embedded AI, and 10 themes for how T&E will need to change for AIES [1].

In order to sufficiently test AIES, the T&E community needs to tackle the following challenges:

- determine testing requirements when state space size makes testing all cases infeasible or the open world problem makes enumerating all cases impossible;
- address the potentially invalid assumption that these emergent systems can be decomposed; and
- deal with dynamically varying systems that are potentially never in a "final" state during deployment [1].

Figure 1 summarizes the 10 different themes outlined to enhance T&E in order to address the challenges with adequately testing and evaluating AIES. Over the past year, Virginia Tech has worked to test and evaluate a variety of AIES. This best practice guide adds further refinement and context to the themes in Figure 1. The best practices contained in this article translate these themes into executable T&E practices. In developing the guide, we leverage our experience working in T&E for both AI systems development and our work with the wider AI community. The best practices captured here reflect an initial attempt to make T&E for AIES tractable. These practices need to be tested against a variety of AIES to ensure they are truly best practices. One highlight carried through many of the best practices is the important role of data. Data is no longer just a product of T&E. It is now an input to the development of the AI system itself. This notable change drives new requirements and practices for T&E of AIES. Additionally, this list is far from complete and should be seen as a living documentation of practices. As more AI systems become available for testing, new practices will evolve and this list will need to be updated. However, each of the practices in this document has proven useful in testing DoD AIES.

Figure 1: Themes for enhancing T&E to address AI challenges adapted from [1]

T&E BEST PRACTICES TO ADDRESS AI CHALLENGES

BEST PRACTICE 1: AIES require new measures for evaluation.

As mentioned in Freeman's 2020 article, AIES still require legacy evaluation measures for effectiveness, suitability, and survivability. However, these measures may have slightly different interpretations based on the implementation of the AI within the system. Additionally, AIES require new measures. Specifically, two new areas of measurement are: 1.) assessment of the AI algorithm or ML model including the data used to train the model; and 2.) assessment of the human-agent team. Figure 2 shows the comprehensive space for measurement and evaluation of AIES highlighting that traditional measures of effectiveness, suitability, and survivability (left) must now be augmented for AIES (right).

A defensible T&E program provides sufficient information to support decision-making in terms of expected performance of the AI algorithm and whether performance changes based on operational environments. For an AIES, this means that we should include the direct measurement of AI algorithm performance. Also, it is necessary to understand if the training data is adequate based on the intended deployed environment, hence the addition of a training data sufficiency measure.

Measures of human-agent teams are also important to consider as non-autonomous AI alone does not accomplish missions, but the AI interacting with human operators does. Arguably, human-system integration has always been a component of the operational suitability evaluation, but the complexity of an AI algorithm's interactions with humans requires that new emphasis with additional measures be considered for AIES.

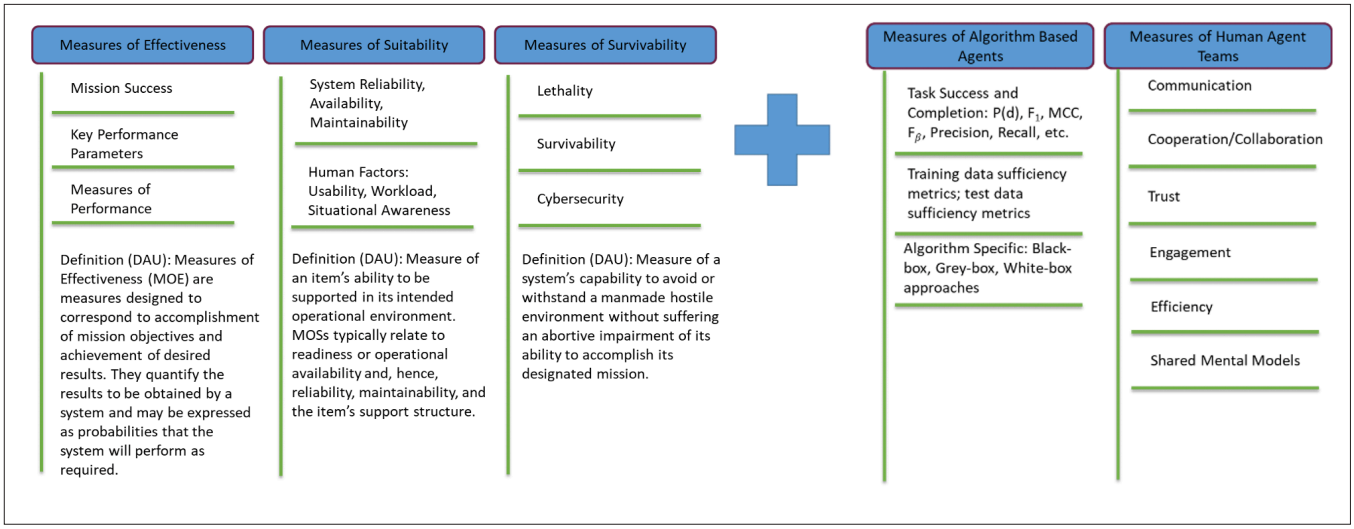

Figure 2: AIES Require an Expanded Set of Evaluation Measures

BEST PRACTICE 2: The T&E continuum for AIES should include data, algorithm, system, and operational testing.

Existing T&E paradigms following the Systems Engineering "Vee" include unit/sub-system testing, integration testing, systems testing, and operational testing. A primary difference worth highlighting between conventional software-enabled systems and systems employing ML is that the software now not only contains programmed algorithms in the traditional sense but also algorithms that are learning from data. This change requires T&E programs to expand unit testing from just algorithm testing to also include training data sufficiency. Additionally, different combinations of training data, learning algorithm, and hyperparameter tuning can result in substantially different models due to interactions, necessitating comprehensive integration testing. An adequate T&E program for an AIES requires that these types of testing are incorporated with corresponding measures in the T&E continuum.

It is important to highlight that AI algorithms and the systems they are embedded within are inherently coupled, so the evaluation of the AI component cannot be divorced from the system and operation. Figure 3 shows the embedded nature of AI components within a system that then must operate in an operational environment (system context) that includes humans. Since humans are considered part of the system, it is critical to capture human factors measures in the context of human-agent teaming in addition to all previous measurements from operational testing. The measurements are important, but it is also necessary to design, develop, and carry out new experimental methods for capturing the operational impact of AI (see best practice 7).

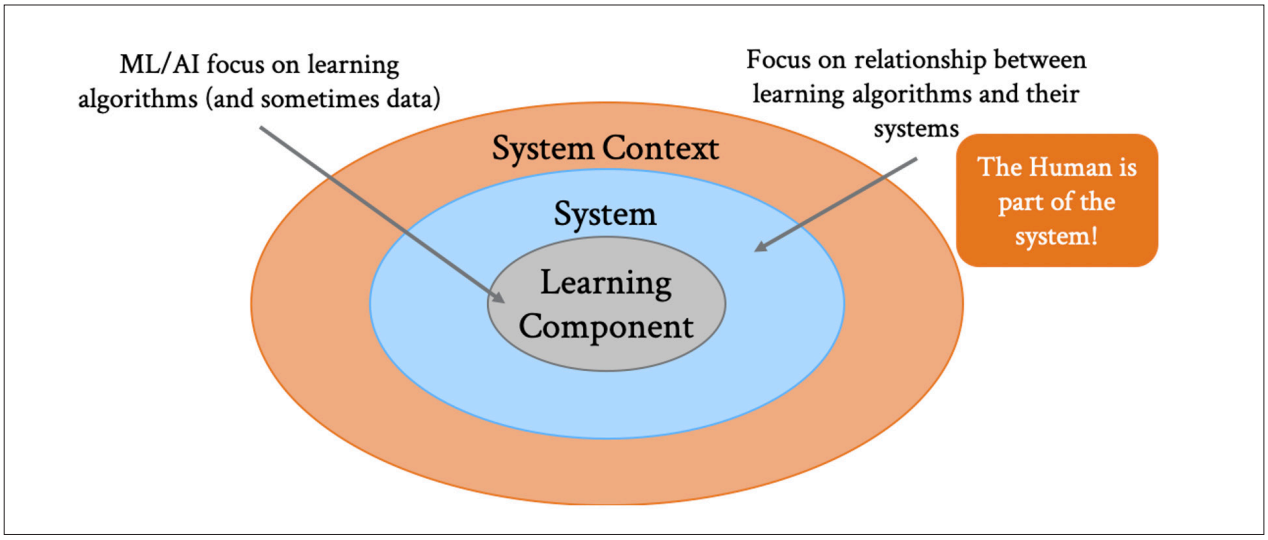

Figure 3: AI and System Levels of Abstraction adapted from [2]

BEST PRACTICE 3: Experimental Design provides methods for efficiently testing AIES across the continuum.

The field of Design of Experiments (DOE) has proven useful in providing a defensible basis for T&E of DoD systems prior to the incorporation of ML and AI. DOE in its broadest definition is a test or series of tests in which purposeful changes are made to the input variables in order to elicit a change in response. Freeman and Warner show the value of statistical thinking in DoD T&E for systems with complex architectures, software, human interactions, and military missions [3]. All of those complexity challenges remain applicable for AIES. Lanus et al. provide a T&E framework for multi-agent systems of intelligent agents and show how different DOE techniques apply depending on the type of testing [4]. Table 1 expands on the work of Lanus et al. to show how new levels of testing (best practice 2) are needed for AIES by adding the requirement for characterization of training data sufficiency. Table 1 provides a quick reference for how different DOE methods apply at different phases in the T&E continuum.

Test Article Focus	Type of Testing	Common Applicable DOE Techniques
Training Data	Unit	Combinatorial Coverage Characterization, Data Latent Space Characterization
AI Algorithm	Subsystem/Unit Testing	Combinatorial Interaction Testing, Optimal Learning
System	Integration Testing, Acceptance Testing, & Developmental Testing	Classic DOE (e.g., factorial designs, response surface designs, optimal designs)
System of Systems (to include humans & operational environment)	Operational Testing	Human Factors Designs, Classic DOE, Bayesian Optimal Designs

Table 1: Applicable DOE Techniques for Varying Levels of Test Articles

BEST PRACTICE 4: Data coverage measures are a component of test adequacy.

As highlighted above, ML requires not only software, but also data. Therefore, the importance of data sufficiency is critical for adequate test programs and must be measured. Lanus et al. [5] defined a new measure of data sufficiency, combinatorial coverage measure (CCM). CCM uses metadata and/or ML features to provide a human interpretable input domain for the ML algorithm in relation to a defined universe. Expanding on CCM, set difference combinatorial coverage measure (SDCCM) is a directional measure of the difference in covered input space combinations between two data sets defined in the same universe, providing a meaningful distance between the two sets when the measure is computed bidirectionally. One application is to characterize the difference in the operating space between where the model was trained and where the model is deployed [6]. A third measure of data information content comes from Cody et al. called transfer distance [6]. Transfer distance computes the distance in latent space between two data sets. Transfer distance utilizes principal component analysis and Gaussian mixture modeling to compute the distance between sets. Initial research shows that this measure correlates with correct classifications. Transfer distance can be used to determine upper-bounds on the error in new operational environments.

Data coverage measures are critical for not only assessing the relationship between the training data and the intended operational environment but are also necessary for designing adequate test programs. Consider the goal of building test sets that provide representative coverage (i.e., matches the training and validation sets as close as possible) or have hard coverage (i.e., exploits edge cases or are adversarial in nature). Figure 4 shows, conceptually, all possible set relationships of coverage between two data sets and highlights the metrics indicating the ideal representative and hard cases. The ability to measure data set coverage provides a mechanism for systematic selection of test sets for ML. The representative case tests how well the model performs in contexts where it should have learned, meaning that the input space difference between the validation set and test set is close to zero. Initial results support that test performance is improved when the difference between the test set and validation set is also close to zero, which suggests both sets are representative of the same operational environment, but this does not provide a measure of generalizability. The hard case tests how well the model generalizes for contexts it has not seen before, and here it is preferred that the difference between the test set and validation set are both close to one.

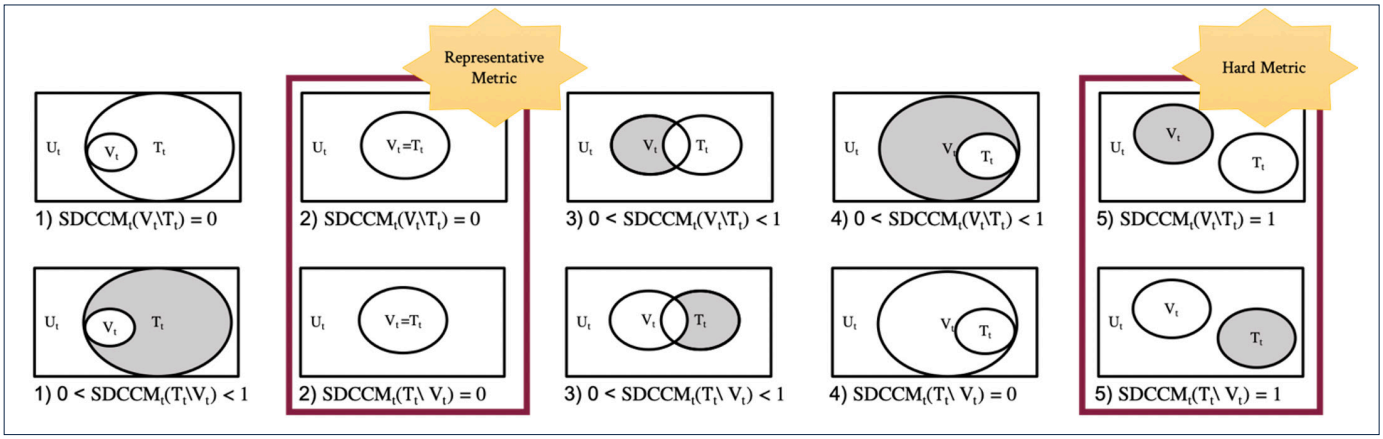

Figure 4: Relationships of Different Set Difference Combinatorial Coverage Metrics and Metric Values Corresponding to the Ideal Representative Case and the Ideal Hard Case. Here V is the Validation Set, and T is the Test Set.

Coverage metrics assist in achieving explainable AI by illuminating the gaps in the input space on which the model learned. A challenge in applying coverage metrics is that they require the data set on which the coverage computation is performed to be defined on features meaningful to the model or metadata acting as a surrogate for these features. These features do not necessarily correspond to the operational environment nor are they always easily captured in metadata, creating challenges for interpretability and collecting the necessary data.

BEST PRACTICE 5: T&E programs for AIES must learn the critical factors for each level of T&E.

Figure 5 shows potential variables that CCM can be calculated across. Metadata are those variables that reflect the conditions under which the data used by the ML model was collected (e.g., UAV altitude, time of day, sensor power, etc.); human interpretable operational factors are those variables that reflect the operating environment (e.g., time of day, weather, operational mission) but may or may not impact the performance of the model; finally, meaningful ML features are those features that actually explain why performance differences exist in the model itself. In the beginning of a T&E process, we may know and have access to metadata and operational factors but probably do not know what the meaningful ML features are. A new aspect of T&E for AIES is that we must learn these important features for our CCM to actually capture the sufficiency of the training data with respect to the operating environment and/or characterize the testing data as having representative coverage or hard cases.

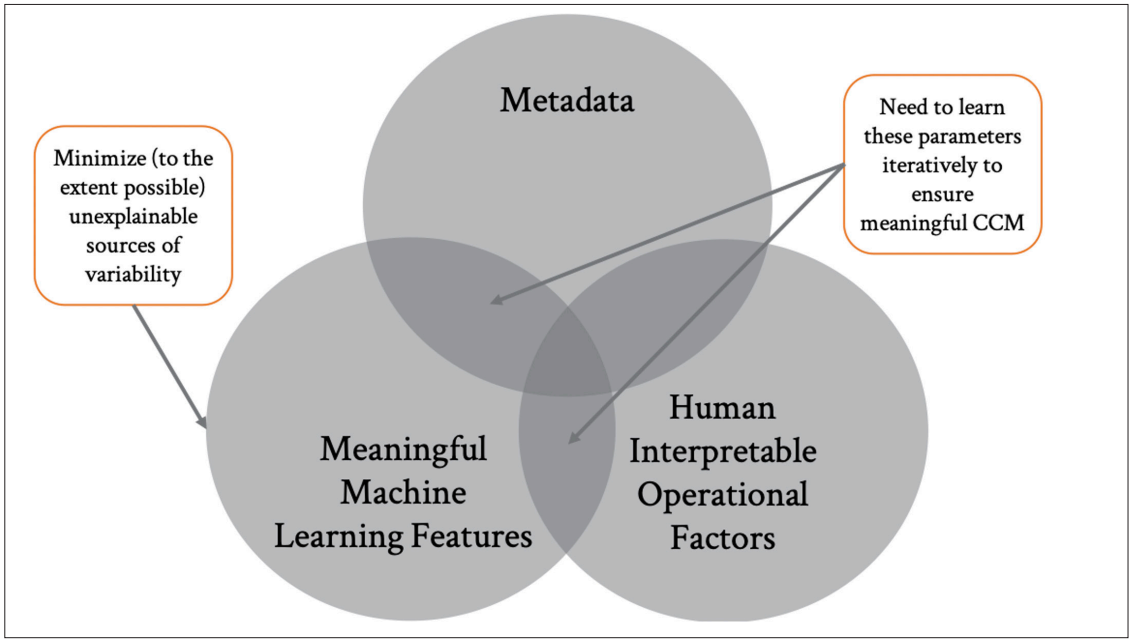

Figure 5: Diagram Highlighting the Relationship Between the ML Features, Operational Factors, and Metadata

BEST PRACTICE 6: T&E programs must include robustness/adversarial test sets.

Adversarial testing has always been an important aspect of T&E to include opposing forces and the systems they will leverage, cybersecurity attacks, and degraded environments that our systems might encounter when facing a peer or near-peer adversary. For AIES, we must also consider how adversaries will capitalize on new vulnerabilities introduced by the AI. One best practice is to test AIES against data and contexts the algorithm has not seen before. This includes developing robust test cases (edge cases) as shown in Figure 4, but can also include attacks targeted toward AI vulnerabilities.

BEST PRACTICE 7: Side-by-side (or head-to-head) operational testing can quantitatively capture the impact of AI in an AIES.

Side-by-side testing of new systems against legacy capabilities is not a new concept. It has been used in the past to show that a new capability surpasses legacy capability. The biggest limitation of side-by-side testing is the cost since it essentially doubles the test requirements. However, for AIES, the cost is worth the investment right now.

Figure 6 shows the concept behind side-by-side testing for AIES. Essentially, Figure 6 shows that an operational test might include defining a set of operational missions and tasks, ensuring that they span key operational factors, and then each resulting test run is assigned to either

only a human and corresponding system (top) or a human plus AIES (bottom). For example, in location monitoring tasks, an analyst might use direct feeds from a UAV, or they could turn on AI-enabled detections and use the combination in conducting their missions.

In a within-subject design, all humans are assigned the missions both with and without AI (so they complete the same mission/task twice), and the assignment is ideally counter balanced so they are randomly alternating between whether they complete the mission with or without AI first. In a between-subject design, the humans are randomly assigned to complete the mission either with or without AI, and a large enough sample of humans is used to ensure that performance differences are due to the AI, not the humans.

Side-by-side testing has lots of benefits for T&E of AI systems. Notably, there is a large perceived risk with deploying AI. Side-by-side testing allows that risk to be put in the context of the risk that was already implicitly assumed by only having humans executing missions. This provides a mathematical basis for decision makers as they contemplate the pros and cons of fielding AIES. Additionally, it provides the opportunity to take a rigorous look at human systems integration issues and human-agent measurement as captured in best practice 1.

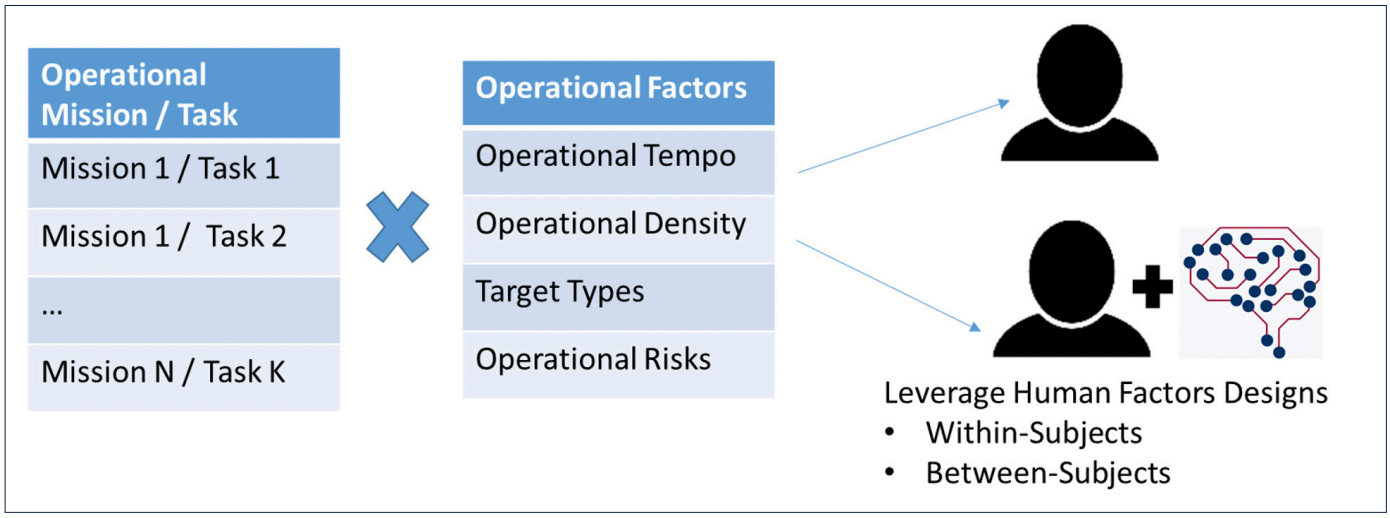

Figure 6: Experimental Concept for Side-by-Side Testing of AIES

BEST PRACTICE 8: Data management is essential.

A theme that prevails across this best practice guide is that data is essential. Building methods that integrate disparate data to be used effectively by AIES requires data processing and management from all potentially relevant data sources. It is necessary to know how to handle the data, and future evaluations may fuse information across multiple phases of testing. Integration of disparate data, or data fusion, provides more mission context and enables a more focused operational environment for the AI system.

CONCLUSIONS

Freeman [1] has documented the challenges of test and evaluation for AIES and the areas that need to be addressed in order to perform T&E of these systems. Ultimately, rigorous, defensible testing of AIES will require new methods and measures. Despite this need, this does not imply starting over, since much of the legacy defensible test methods are still best practices. The new areas of emphasis include:

- training and test data sufficiency from an algorithm perspective;
- human-machine teaming test adequacy; and
- testing across the design, develop, deploy, sustain life cycle addressed for AIES.

These best practices need additional vetting, and numerous practices are surely missing. For example, the research team has not had the opportunity yet to monitor a fielded system using AI. Therefore, none of the practices reflect the need for continuous system monitoring (or online T&E) of a fielded system. We hypothesize that statistical process control on both data coverage metrics and algorithm performance may eventually be captured as a best practice for monitoring the safety and security of AIES, but we have not included that as a best practice in this guide. Most assuredly, other best practices are missing and need to be added to this list.

REFERENCES

[1] L. J. Freeman, "Test and evaluation for artificial intelligence," *Insight*, 23(1), pp. 27-30, 2020.

[2] T. Cody, "Mesarovician Abstract Learning Systems," in *International Conference on Artificial General Intelligence*, Springer, Cham., pp. 55-64, October 2021.

[3] L. J. Freeman and C. Warner, "Informing the Warfighter — Why Statistical Methods Matter in Defense Testing," CHANCE, 31(2), pp. 4-11, 2018.

[4] E. Lanus, I. Hernandez, A. Dachowicz, L. J. Freeman, M. Grande, A. Lang, 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), pp. 203-209, June 2021, IEEE.

[5] 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), pp. 81-84, April 2021, IEEE.

[6] T. Cody, S. Adams, and P. A. Beling, "A systems theoretic perspective on transfer learning," in 2019 *IEEE International Systems Conference* (SysCon), pp. 1-7, April 2019, IEEE.

ABOUT THE AUTHORS

Dr. Laura Freeman is the Director and Research Associate Professor, Intelligent Systems Lab and is the director of the Hume Center for National Security and Technology's Intelligent Systems Lab director. Additionally, she is a research associate professor in the Department of Statistics and a faculty member of the Commonwealth Cyber Initiative.

Her research interests include experimental design considerations in machine learning and artificial intelligence, cybersecurity analytics, reliability analysis, and statistical engineering.

Freeman holds memberships with the National Defense Industrial Association, the American Statistical Association, and the International Test and Evaluation Association (ITEA) as the Editor-In-Chief of the ITEA Journal of Test and Evaluation. She serves in leadership roles within the Systems Engineering Research Center and Acquisition Innovation Research Center as a research council member and as principal investigator in multiple research tasks.

Freeman received her Ph.D. in statistics, an M.S. in statistics, and a B.S. in aerospace engineering, all from Virginia Tech.

Dr. Justin Kauffman is a Research Assistant Professor of the Virginia Tech National Security Institute. His research interests include development of high-fidelity computational models and integrating machine learning and artificial intelligence into physics-based models to better predict phenomena of complex physical systems. Dr. Kauffman has a B.S. in Engineering Science, a B.S. in Mathematics, and a M.S. and Ph.D. in Engineering Science and Mechanics, all from The Pennsylvania State University.

Dr. Erin Lanus is a Research Assistant Professor at the Virginia Tech National Security Institute. Her research applies combinatorial testing to security and AI assurance. She leverages metric and algorithm development for constructing test sets, measuring the quality of data sets, and evaluating systems with embedded AI. Dr. Lanus has a B.A. in Psychology and a Ph.D. in Computer Science, both from Arizona State University.

Mr. Daniel Sobien is a Research Associate at Virginia Tech's National Security Institute in Arlington, VA. His relevant research experience includes image classification and segmentation, data augmentations, testing and evaluation of computer vision models, and analysis of AI and human performers for detection and tracking objects in full motion video. His other research interests include AI assurance and causal machine learning. Mr. Sobien has a BS and MS in Aerospace Engineering, both from Virginia Tech.

Dr. Tyler Cody is an Assistant Research Professor at the Virginia Tech National Security Institute. His research interest is in developing principles and best practices for the systems engineering of machine learning and artificial intelligence. He received his Ph.D. in systems engineering from the University of Virginia in May 2021 for his work on a systems theory of transfer learning.

DISCLAIMER

The Acquisition Innovation Research Center is a multi-university partnership led and managed by the Stevens Institute of Technology and sponsored by the U.S. Department of Defense (DoD) through the Systems Engineering Research Center (SERC)—a DoD University-Affiliated Research Center (UARC).

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)) and the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under Contract HQ0034-19-D-0003.

The views, findings, conclusions, and recommendations expressed in this material are solely those of the authors and do not necessarily reflect the views or positions of the United States Government (including the DoD and any government personnel) the Virginia Tech National Security Institute and the Stevens Institute of Technology.

No Warranty.

This Material is furnished on an "as-is" basis. The Virginia Tech National Security Institute and the Stevens Institute of Technology make no warranties of any kind—either expressed or implied—as to any matter, including (but not limited to) warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material.

The Virginia Tech National Security Institute and the Stevens Institute of Technology do not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.