Developmental Test and Evaluation of Artificial Intelligence-Enabled Systems Guidebook

February 2025

Office of the Director,
Developmental Test, Evaluation, and Assessments

Office of the Under Secretary of Defense
for Research and Engineering

Washington, DC

Developmental Test and Evaluation of Artificial Intelligence-Enabled Systems Guidebook

Office of the Director, Developmental Test, Evaluation, and Assessments
Office of the Under Secretary of Defense for Research and Engineering
3030 Defense Pentagon
Washington, DC 20301
osd.r-e.comm@mail.mil
https://www.cto.mil/dtea/

Executive Summary

The Department of Defense (DoD) developed this guidebook to support the developmental test and evaluation (DT&E) of artificial intelligence (AI) systems and AI-enabled systems (AIES). Its intent is to provide technically sound, consensus-based guidance designed to address the unique challenges posed by AI technologies. The guidebook aims to support government test teams in planning and executing DT&E for AI-enabled components, applications, and systems, while assisting in delivering critical insights to decision makers and stakeholders during AIES development and deployment. Recognizing that AI is a rapidly evolving field, this guidebook reflects the T&E community's current consensus and will likely require updates as technology and methodologies advance.

Testing AI systems presents a key challenge because traditional comprehensive testing approaches are no longer feasible for many AI components because of factors such as the following:

- Inherent unpredictability of model outputs in practice.
- Model sensitivity to small changes in input.
- Complexity and opacity of some AI models.
- High dimensionality of parameter spaces.
- Complex dependence of model output on training datasets.

Furthermore, the normally rapid pace of configuration changes of systems under consideration adds another layer of complexity to the T&E process. These factors undermine the ability of test teams, evaluators, and executives to generalize results from particular tests to support the necessary evaluations of AI components and AIES for engineering or acquisition decisions.

To address these challenges, the guidebook emphasizes several new approaches:

- Early Involvement in Development. Involving T&E teams early in AIES development enables mission-informed technology characterization. This early involvement is essential given the iterative nature of machine learning model development: From the beginning of development, continuous refinements require ongoing evaluation to ensure that the fielded system aligns with operational goals.
- Formal Methods for Augmentation. Formal methods offer mathematically rigorous techniques that complement traditional physical testing, allowing for more precise validation of AI systems. These methods help address the inherent complexities and uncertainties associated with AI technologies.

- Ensuring Testable Requirements. The DT&E community has traditionally collaborated with the requirements community to ensure that system requirements are testable. The complexity of testing AIES expands this role. Efforts focus on ensuring not only the testability in principle but also the ability to develop a viable test program to support the necessary evaluations.
- Informing System and Concept of Employment (CONEMP) Development. The iterative nature of AIES development and its close coupling with CONEMPs require DT&E measurement activities that inform system and CONEMP developers. Testing in areas such as human-systems integration, calibrated trust, emergent behavior, and human-machine teaming and the adherence to responsible AI policies will be critical to avoid costly rework and ensure alignment between system design and operational needs.

This guidebook ultimately aims to serve as a valuable resource for DoD AI efforts, enhancing the DoD capability to effectively test and evaluate AI technologies and ensure their successful integration in support of national defense.

COLLINS.CHRISTOPH
ER.CLAY.1050967561

Digitally signed by
COLLINS.CHRISTOPHER.CLAY.105
0967561
Date: 2025.02.27 12:53:56 -05'00'

Mr. Christopher C. Collins
Director, Developmental Test, Evaluation, and
Assessments

Contents

Figures

Tables

1 Introduction

This guidebook is intended for government test teams planning and executing developmental test and evaluation (DT&E) of artificial intelligence (AI)-enabled systems (AIES) or AI components to be used in systems. It provides focused guidance for pre-program and program acquisition DT&E activities involving AI.

The presence of AI introduces major changes to DT&E. Machine learning (ML) approaches and responsible AI (RAI) mandates alter traditional approaches to system and software life cycles, including characterizing performance and risk. Test and evaluation (T&E) must be pervasive throughout the entire AIES development to be effective and convincing. This pervasiveness often requires substantial DT&E engagement with science and technology (S&T), prototyping, and experimentation efforts.

This initial guidebook release addresses how AI changes DT&E. It examines the changes from the complementary viewpoints of DT&E activities and outputs and the AI drivers of changes. The guidebook covers ML datasets and includes how to test and evaluate the datasets, the models trained using them, and the systems of which they are a part. The guidebook also provides a brief discussion of potential benefits across the enterprise related to expanded interactions between the T&E community and other communities of practice. These expanded interactions are crucial for the early involvement of the DT&E community in defining an effective and suitable concept of employment (CONEMP) for the system.

The first release of this guidebook introduces and discusses the following subject matter:

- **Section 1 – Introduction**: Discusses the purpose and scope of the guidebook.
- **Section 2 – DT&E of AIES Overview**: Examines the recent advances in AI systems that have implications for DT&E responsibilities in performance evaluation, risk assessment, and support to systems engineering. The issues raised by AI, and in particular by ML, are introduced at a high level.
- **Section 3 – AI-Driven Changes in T&E Practice**: Presents specific T&E methodologies relevant to the novel challenges of ML.
- **Section 4 – Expanded Interactions for the T&E Community**: Addresses expanded organizational and professional interactions outside the T&E career field.

Future releases of this guidebook will expand on the topics addressed in this release, including emerging issues associated with T&E of generative AI and a more complete discussion of reinforcement learning (RL), as needed to address emerging Department of Defense (DoD) use cases. Future process-oriented content will include a more detailed discussion of the T&E role in risk management and safety engineering as well as the verification, validation, and accreditation (VV&A) of data and models, as DoD and its Components expand their policies and standards in those areas.

1.1 developmental Test and Evaluation as a Continuum

The dTEaaC framework leverages the principles of Agile and scales them to move T&E holistically from a serial set of activities to an integrative framework focused on a continuum of capability and outcome-focused testing, Agile scalable evaluation, and enhanced test design facilitating an ongoing campaign of learning.

This guidebook is part of a broader set of T&E guidance developed to support the implementation of a paradigm shift into dTEaaC. Key enablers include model-based systems engineering (MBSE) and digital engineering (DE) (authoritative source of truth); incorporation of technological innovations; a supportive infrastructure; and a transformed culture.

Each key enabler plays a critical role in conducting dTEaaC. Combined, the key enablers will provide timely and comprehensive information on system performance and risk characterization from the earliest design stages, enabling rapid development and fielding along with ongoing support for these increasingly complex systems and SoS. T&E, as an integral part of SE and ME processes (facilitated by using DE), not only supports decision making on individual systems but also helps enable the informed management of DoD capability development portfolios.

The three key tenets and multiple key attributes and enablers for implementing dTEaaC are as follows:

Key Tenet 1: Links a Campaign of Learning to Warfighter Needs across the Capability Life Cycle

Establishes a Campaign of Learning based on data, information, and knowledge across the capability life cycle.

Promotes continuous mission validation via an Agile, iterative, data-driven, and knowledge-based framework for capability delivery, moving away from an event-based, serial approach to a data-driven approach.

Facilitates critical testing and validation of emerging technologies, including artificial intelligence (AI)-enabled and software-intensive systems.

Optimizes S&T, P&E, and test execution in support of pre-acquisition technology maturation, readiness, transition, and risk reduction.

Emphasizes the assessment of warfighting capabilities across the testing continuum.

Supports continuous validation of fielded Agile development and learning systems.

Key Tenet 2: Provides Timely Decision Support

Promotes a decision framework that integrates insights from T&E early in the life cycle in support of complex decision making, ensuring an enhanced level of validation of warfighting needs through the alignment of data, analysis, and evaluation to support capability delivery decisions. This decision framework leverages the Integrated Decision Support Key (IDSK) framework applied both pre-acquisition and post-acquisition, integrating into an enterprise-level integrated data framework.

Enables earlier discovery and mission-focused decision making.

Provides ongoing and continuing learning and feedback to build better understanding for decision makers.

Enables near real-time decision dashboards and visualizations directly supported by the specific relevant authoritative data.

Key Tenet 3: Aligns T&E with the Digital Innovation Ecosystem

Fully integrates and aligns T&E with model-based DE development and application, ensuring an integrative environment between ME, SE, and T&E models and processes across the capability life cycle.

Closely integrates with ongoing digital workforce initiatives and curriculum to develop the requisite T&E engineering skills and experiences for the workforce.

Builds the foundation for a modeling and simulation (M&S) “digital thread” continuum documenting the core testing required to validate and accredit models.

Leverages the modeling environment to complement and improve live testing.

Establishes the foundational integrated data and information architecture that is essential for seamless knowledge sharing and enhanced decision making.

2 DT&E of AI-Enabled Systems Overview

2.1 AI-Enabled Systems

AI technologies have the potential to be a significant enabler of advanced capabilities and a key part of the National Defense Strategy. The 2023 DoD Data, Analytics, and AI Adoption Strategy states that “accelerating adoption of … AI technologies will enable enduring decision advantage, allowing DoD leaders to prioritize investments to strengthen deterrence; link cross-cutting campaign outcomes that counter our competitors’ coercive measures; and deploy continuous advancements in technological capabilities to creatively address complex national security challenges.” The strategy goes on to state that “responsible AI [is] the Department’s dynamic approach to the design, development, deployment, and use of AI capabilities in accordance with the DoD AI Ethical Principles while delivering better, faster insights and improved mission outcomes” and that “sound assurance processes for testing, evaluation, validation, and verification are imperative for Responsible AI.”

AI is in the news every day, but it is remarkably difficult to define exactly what AI is. This difficulty is partly because AI is an extremely multidisciplinary field, and all the different research communities that contribute to AI have their own unique jargon and conventions. In addition, both the technical literature and the popular concept of AI keep evolving—many things that would have been considered AI 20 years ago are now so commonplace that they are just thought of as “software” or “apps.”

In discussions of how AI affects the characterization of system capabilities, limitations, and risks, the guidebook focuses on the specific AI technologies and methods that pose novel challenges to T&E activities. In discussions of T&E’s role in supporting the DoD AI Ethical Principles and RAI mandates, the guidebook takes a broader view that applies to any complex software-enabled functions.

2.1.1 Procedural AI

This guidebook is not focused on DT&E of systems whose only AI is procedural AI. The March 28, 2024, Director of the Office of Management and Budget (OMB) Memorandum, “Advancing Governance, Innovation, and Risk Management for Agency Use of Artificial Intelligence,” explicitly does not address “ robotic process automation or other systems whose behavior is defined only by human-defined rules or that learn solely by repeating an observed practice exactly as it was conducted.”

For many years, the vast majority of AI research has involved computer programs written in human-readable programming languages to implement human-understandable algorithms. The products of this research include expert systems; heuristic optimization and search techniques; robotic process automation; theorem-proving and reasoning systems; and conventional grammar and spelling checkers. This kind of AI goes by various names—procedural AI, symbolic AI, “good old-fashioned AI,” first-wave AI, and so forth (e.g., McAdams 2021).

From a T&E perspective, nothing is new or special about procedural AI. It is “just software,” with all the challenges that testing software brings. Standard software testing methods apply, and no unique challenges exist in translating behaviors and specifications from the component level to the system level. The DoD AI Ethical Principles certainly apply to systems that use procedural AI, but the T&E tools and methods needed to support the procedural AI adherence to those principles are best understood in a general context of software T&E, rather than as AI-specific challenges.

2.1.2 Machine Learning

This century has seen rapid progress in a new kind of nonprocedural AI called machine learning (ML). The idea behind ML is this: Rather than designing a procedure to calculate the best answer, developers apply a process that lets a model *learn* a procedure by examining a large number of examples, with corrective feedback for incorrect outputs. (Note: This is a different use of “model” than in the context of “modeling and simulation,” and the distinction is important.) The mathematics of ML is similar to familiar statistical methods such as classical regression or Bayesian estimation, but the models that are learned typically have many more parameters than traditional statistical models. In addition, unlike traditional regression models, those parameters generally do not have straightforward interpretations in terms of the inputs and outputs of the problem they are trying to solve. The large number of parameters and the absence of straightforward interpretations challenge traditional test practices.

Some ML models, such as those used in large language models (LLMs) (see Section 2.1.3), may have billions of parameters. Computer-vision models may have hundreds of millions of parameters (Tulbure et al. 2022). It is convenient to divide the universe of ML models into three main types:

- **Unsupervised learning (UL) models**. UL models discover patterns in data without the use of ground truth information about how the data elements should be grouped or which group each element might belong to. Common UL uses include clustering, distribution fitting, anomaly detection, and text mining. Social network analysis, topic modeling, and fraud detection are examples of current applications of UL (e.g., Celebi and Aydin 2016).

- **Supervised (and semi-supervised) learning (SL) models**. SL models use ground-truth labels associated with the training data to learn how to predict the appropriate labels for unlabeled data. Common SL uses are classification and prediction, each of which has many potential applications. Facial recognition, cancer screening, and speech-to-text processing are all examples of current tasks using SL. Semi-supervised learning, as the name implies, takes advantage of having a relatively small number of labeled data instances to improve the performance of UL algorithms developed on the larger set of unlabeled data (e.g., Igual and Seguí 2024).
- **Reinforcement learning (RL) models**. RL is a technique that applies in contexts where the model must make a sequence of choices in a changing (and perhaps adversarial) environment to achieve a goal. Unlike SL, where for each input there is a known correct output, in RL, the "correctness" of individual output choices can be determined only at the end of the task, according to how well the model has succeeded in its goal. The trained model describes a policy or strategy that specifies what action to take at any time, given the available information about the situation. The size of the trained model depends on the number of possible situations (the state space) and the number of possible actions in each state. RL has famously been used to train models to play games such as chess and gō as proficiently as the best human players. It is also used for a wide range of other applications, such as autonomous vehicle navigation, robot motion controller design, automated defensive weapon systems, electronic warfare systems, and active power management (e.g., Sutton and Barto 2018).

ML models may also be used in combination. An SL model might be used to generate synthetic test data for a UL model. Both SL and RL often use an approach called "generative adversarial networks" in which two models are trained simultaneously: one model trying to solve the task of interest and a separate model attempting to find ways to thwart the first model. For example, if the primary model is trying to learn to recognize specific faces, the second model would try to learn to generate faces that the first model would incorrectly classify (e.g., Creswell et al. 2018).

From a T&E perspective, ML is the kind of AI that may require changes to test strategy, planning, and execution over the development and fielding life cycle. ML depends on its training process in ways that introduce risks different from those found in traditional algorithmic software. In addition, ML models can be more sensitive to small changes in input or environment and exhibit more extreme worst-case behavior, relative to traditional software. Diagnosis of ML defects and characterization of ML capabilities and limitations may require novel instrumentation and test designs. This guidebook will focus on the novel T&E implications of ML.

2.1.3 Generative AI

Generative AI (abbreviated GenAI or GAI) is ML that appears to mimic human outputs in various media (also called *modalities*). Widely used GenAI models include the ChatGPT family of language output models developed by OpenAI; the Llama family of language output models developed by Meta AI; and the DALL-E family of image output models developed by OpenAI.

GenAI typically takes inputs in the form of text strings called *prompts* that request or describe the desired output. For example, prompts to a system that produces text output might take the form of questions or directions such as:

- "Write a sonnet in the style of William Shakespeare on the subject of cheeseburgers."
- "What are the main differences between Indian and Thai cuisines?"
- "Summarize the plot of *Moby-Dick* in bullet points."
- "Write a JavaScript program to invert a square matrix of floating point numbers."

Prompts to a system that produces output in the form of images might include:

- "An anime-style cartoon of a tiger playing tennis with a raccoon."
- "A pen-and-ink sketch of a man with a long beard and bushy eyebrows."
- "A photo of geese landing on a pond in winter."

The models used in GenAI applications are often extremely large and complicated and usually involve several submodels trained to do supporting functions. They typically include an LLM trained on an enormous set of documents and may make use of any or all of SL, UL, and RL techniques. The largest current version of the Llama LLM, for example, has 70 billion parameters.

Unlike typical SL, UL, and RL applications, GenAI models are often used for purposes that are quite different from what they (or their component models) were explicitly trained to do. For example, the ability of ChatGPT to write computer programs given high-level functional descriptions was something of a surprise to its developers. Such *emergent capabilities* are part of the exciting promise of GenAI as well as a significant challenge for T&E, which may need to evaluate not only how well a model works for a specified task but also which other tasks it is (or is not) reliable for. T&E of GenAI is still in its infancy as a discipline and is an active area of both academic research and DoD investment (e.g., Banh and Strobel 2023).

2.1.4 Key Recurring Challenges for T&E of AI

The intent of this guidebook is to alert DT&E personnel to the specific challenges and risks associated with AIES, including challenges associated with ensuring compliance with DoD policies regarding ethical and responsible use of AI. Because the novel challenges and risks are almost entirely driven by the use of ML, this guidebook will focus on ML-specific risks. These challenges include:

- Unpredictability of model outputs in practice.
- Model sensitivity to small changes in input.
- Complexity and opacity of some models.
- High dimensionality of parameter spaces.
- Complex dependence on training datasets.

2.1.5 A Note on AI vs. Autonomy

These design engineering choices are important for T&E planning because changes in system design during prototyping and development may alter how or whether AI is being used, which in turn may affect test design and instrumentation needs.

The terms “autonomy” and “AI” are sometimes used interchangeably in casual discussion, but they are two distinct concepts. A useful starting point to compare and contrast DT&E of AI and DT&E of autonomous systems (or autonomy) is provided by the AI Acquisition Guidebook: “The following distinction between autonomy and AI should be recognized – autonomy refers to an agent or machine being delegated to perform a task, while AI is a means to achieve that goal.”

Autonomy refers to an agent or machine being delegated to perform a task—capable of independent operation without external control. This capability could be mission essential and therefore a requirement of the program. It could also be broad or narrow—for example, a requirement for general autonomous navigation versus a requirement to be able to parallel park without human intervention.

AI methods might be used to achieve autonomous capabilities. Whether to use AI and what kind of AI model to use are design engineering choices, not system requirements. In practice, certain capabilities (such as rapid automated language translation) can currently only be accomplished using AI, but these are the exception, not the norm.

DoD made a conscious decision to publish a separate guidebook, the forthcoming DT&E of Autonomous Systems Guidebook, that will provide a more complete discussion of the unique T&E challenges of autonomous systems. DoD will consider a combined T&E of AI and Autonomous Systems guidebook in a future release.

2.2 DT&E Activities and Outputs

The forthcoming DoD Instruction (DoDI) 5000.DT, "Test and Evaluation," calls out several a number of explicit purposes of DT&E. These purposes can be grouped under four main headings, each of which encompasses multiple T&E activities and outputs:

- Characterizing performance.
- Characterizing risk.
- Informing systems engineering.
- Informing acquisition management.

The following subsections expand on these purposes highlighting DT&E activities and outputs affected by the use of AI.

2.2.1 Characterizing Performance

Characterizing system performance is a core purpose of T&E, beginning with early designs and prototypes. At various points in the development cycle, characterizations are required at the component, subsystem, full-up system, and mission (system-of-systems) level. DT&E is specifically concerned with:

- Characterizing AIES capabilities.
- Characterizing AIES limitations.
- Characterizing defects and deficiencies.
- Determining performance envelopes.

2.2.1.1 Characterizing AI-Enabled System Capabilities

At the component level, assessment of AI model performance may involve specialized techniques and metrics. This is especially true of ML models, including classifiers, regression models, UL models, RL models, and GenAI models. These models are discussed in more detail in Section 3.2.4.

An important aspect of AI-enabled systems or subsystems is that their overall performance is often not well characterized by average or typical system performance. For many ML approaches, deviations from intended behavior do not obey a predictable statistical distribution and do not degrade smoothly at the boundaries of some well-defined performance envelope. Some ML models also exhibit *brittleness*, in that very small changes in operational inputs can lead to sharply different ML model outputs. This brittleness results in practical irreproducibility in testing because it may not be possible to exactly reproduce test conditions at the model-input level.

As a result, ML-enabled systems (MLES) will typically require separate characterization of average and worst-case performance. These characterizations will potentially require different test methodologies. For example, it might be possible to characterize average system performance using traditional design of experiments (DOE), whereas characterizing worst-case performance requires sequential red-teaming techniques intended to find failure modes lurking between the design points of a traditional test plan.

MLES can continue to learn and change during operation. DT&E activities must account for a system’s performance characteristics varying over time. In the presence of ongoing learning, regression testing will need to address worst-case as well as average performance.

2.2.1.2 Characterizing AI-Enabled System Limitations

As noted in Section 2.2.1.1, DT&E of AIES will typically need to focus extra attention on potential unacceptable or worst-case performance, in addition to characterizing average or typical performance. Standard statistical techniques, such as DOE, covering arrays and response surface methods, may not be sufficient to accurately model the range of system behaviors. ML models, unlike physics-based models or standard statistical models, do not always exhibit smooth and bounded output responses over typical input ranges. DOE methods require interpolation of model response between the tested design points, and the methods assume that this response will be continuous and that interpolation errors can be bounded, based on the observed variance at the test design points. DOE methods thus tend to overestimate the predictability of MLES behavior and fail to identify worst-case behaviors.

Search-based adversarial testing methods may be useful for finding specific model inputs or situations for which model performance is undependable. In some cases, formal methods can also be applied to trained models to compute bounds on their worst-case errors (see Section 3.4).

An important consideration in MLES is the validity of the training data used to develop the model, relative to the intended operational environments and missions. Identification of coverage limitations of the training data is part of the data VV&A process. DT&E can support program managers (PMs) in identifying shortfalls in training data correctness or coverage.

2.2.1.3 Characterizing Defects and Deficiencies

Some limitations are sufficiently critical that they are considered defects or deficiencies in the system. DT&E supports identification and diagnosis of these deficiencies. Explainable AI (XAI) techniques (see Section 3.5) can sometimes provide insight into when and why the system does not perform as intended. For RL models, a key DT&E function is to detect and diagnose “reward hacking” behaviors. Reward hacking is discussed in more detail in Section 3.6.3.

As with characterizing AIES limitations (Section 2.2.1.2), search-based adversarial testing methods may be needed to efficiently find and characterize system deficiencies. Some of these methods are best implemented as sequential test designs, where the outputs of each test inform the inputs to the next test. Several tools have been developed to support this kind of automated, iterative, adversarial testing of AIES. All of these tools assume that an accredited simulated test environment is available (e.g., Soklaski et al. 2022).

2.2.1.4 Determining Performance Envelopes

For physical systems, the performance envelope is typically well-defined: It can be described fairly simply in terms of limits on environmental conditions and mission tasks as well as system performance. Within that envelope, behavior generally changes smoothly and relatively slowly. For an aircraft, the performance envelope might be stated in terms of an altitude range, a cruising speed range, and permissible weather conditions. For an indirect-fire weapon, it might be stated in terms of minimum and maximum range to target, required precision, and maximum rate of fire.

Equally important, the exceptions to smooth behavior tend to be well understood. For example, it is known from physics that velocities near Mach 1 for aircraft and artillery can cause rapid changes in performance that are anything but smooth. Similarly, it is known from human factors research that two-way voice communications suffer sharp suitability degradation as end-to-end latency approaches 200 milliseconds.

For many commonly used ML techniques, small changes in model inputs can lead to sudden significant changes in model outputs. This sensitivity can manifest as brittleness (described in Section 2.2.1.1); it can also lead to a situation where the boundaries of acceptable system performance are not “edge cases” in the usual sense but instead occur intermittently throughout the parameter space of interest. In most cases involving ML, no physical or empirical theory exists that can predict where the analogs of Mach 1 and 200 milliseconds will be. Characterizing model and system performance envelopes under these conditions imposes new challenges in identifying problematic regions in the system parameter space and in concisely describing these regions to operators and decision makers.

2.2.2 Characterizing Risk

Informing PMs and other stakeholders regarding system risks is another key function of DT&E. Leaders cannot make good trade-offs between capability and risk unless they are well-informed about both. For AIES, the nature and magnitude of risks can be different than for non-AI systems. The National Institute of Standards and Technology (NIST) AI Risk Management Framework (RMF) 1.0, NIST AI 100-1, provides a structured approach to addressing the novel risks introduced by ML; also see the NIST AI RMF: Generative Artificial Intelligence Profile NIST AI 600-1.

The three principal T&E-related elements of the RMF core are map, measure, and manage:

- **Map** – Establish the mission context to assess risk.
    - Identify potential hazards to be guarded against.
    - Document developer choices regarding training data and model selection.
    - Map model performance metrics to mission performance metrics.
    - Identify instrumentation needs.
- **Measure** – Assess, analyze, and track the identified risks.
    - Characterize ML model and system capabilities, limitations, and hazards.
- **Manage** – Prioritize and mitigate the identified and quantified risks.
    - Report T&E evaluations to inform stakeholders regarding risk-benefit trade-offs.

Of particular importance to AIES is the characterization of risks associated with:

- RAI mandates.
- System robustness.
- Technology.

The following subsections provide information on RAI risks, the robustness of MLES, and technology risk characterization for AIES.

2.2.2.1 Responsible AI Risks

DoD policy establishes several a number of specific goals for the acceptable employment of AI-enabled capabilities (AIECs), encompassing but also expanding upon the established goals of operational effectiveness, suitability, and survivability.

Table 2-1 lists the key policy documents at the time of this publicationas of the time of writing. The policies differ in terminology and applicability, but taken together, they call for RAI—that is, the assurance that through responsible development and deployment processes, DoD AIES will have well-defined uses for which the AIES are:

- Safe
- Secure
- Effective
- Governable
- Accountable
- Traceable
- Reliable

Table 2-1: Key Responsible AI Policy Documents

Policy	Date	Source
DoD AI Ethical Principles	2020	DoD
DoD Responsible AI Strategy and Implementation Pathway	2022	DoD
DoD Data, Analytics, and AI Adoption Strategy	2023	DoD
DoD Directive 3000.09, “Autonomy in Weapon Systems”	2023	DoD
Political Declaration on Responsible Military Use of Artificial Intelligence and Autonomy	2023	Department of State
DoD Instruction 5000.61, “DoD Modeling and Simulation Verification, Validation, and Accreditation”	2024	DoD
Director of the Office of Management and Budget Memorandum, “Advancing Governance, Innovation, and Risk Management for Agency Use of Artificial Intelligence”	2024	OMB

Safety and security are standard certifications supported by T&E. Risks associated with system safety are addressed in Section 4.6.7. AI-specific cyber risks will be addressed in the forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook. Effectiveness of AIES is addressed in Section 3.2.43.2.4. Accountability for employment of AIES is not a T&E function, though DT&E characterizations of risks and limitations (see Section 2.2.2) can help leaders make informed choices regarding system employment. DT&E roles with regard to the remaining RAI goals are discussed in the remainder of this subsection.

Traceable AI-Enabled Systems

The DoDAI Ethical Principles require that AIECs be *traceable*—i.e., that relevant personnel possess an appropriate understanding of the technology and development processes used. As stated in the May 26, 2021, Deputy Secretary of Defense (DepSecDef) Memorandum, "Implementing Responsible Artificial Intelligence in the Department of Defense," this understanding includes "technology, development processes, and operational methods applicable to AI capabilities, including transparent and auditable methodologies, data sources, and design procedure and documentation."

A best practice for supporting traceability for ML is the use of model cards and data cards. These cards are artifacts produced by the ML development teams. Model cards describe the type of model employed, its intended functions and operational uses, and any known shortcomings or risks. Data cards give an overview of the data sources and data processing sequence used to develop the training data that produced the deployed model. Model cards and data cards are both useful inputs to test strategy and planning, as they identify potential risk areas that testers may need to address; see Section 2.2.3.3.

XAI techniques can also support traceability by helping users and testers understand the model's behavioral drivers (e.g., see Dwivedi 2023 and the March 28, 2024, Director of OMB Memorandum). Additional details on XAI methods are discussed in Section 3.5.3.

Evaluating the reliability of systems under test is a well-established role of DT&E. For software-intensive systems in general, and for MLES in particular, reliability modeling and evaluation will typically be more complicated than simple mean-time-between-failures tracking and projection. The May 26, 2021, DepSecDef Memorandum clearly calls for an expanded sense of reliability than the mean-interval-between-failure approach. Software does not wear out, and the discovery of software defects does not conform to traditional statistical failure rate models. Errors in ML outputs are similarly unpredictable, both in timing and in magnitude, and will require defining new metrics for model and system reliability. As a result, characterizing the reliability of MLES may involve collecting, evaluating, and conveying more information to decision makers than has been standard practice in the past. Information on the entire history of reliability (and other performance attributes) will need to be retained to support this characterization (e.g., Hong et al. 2022).

Governable AI-Enabled Systems

An important aspect of Federal and DoD policy for AIES is that they be *governable*—i.e., that commanders are able to employ AIES with high confidence that the systems will perform as intended and avoid undesired actions or behaviors. In support of this goal, several specific requirements for AIES have been established:

- AIES should be engineered such that employing forces have the ability to detect and avoid unintended consequences and to disengage or deactivate deployed systems that demonstrate unintended and undesirable behavior.
- Military use of AI capabilities needs to be accountable within a responsible human chain of command and control.
- For autonomous weapon systems (AWS), DoD Directive (DoDD) 3000.09, “Autonomy in Weapon Systems,” establishes additional substantive review processes, including T&E requirements, intended to ensure that AWS avoid unintended engagements; avoid loss of effective control; and allow commanders and operators to exercise appropriate levels of human judgment over the use of force.

The requirements to be able to detect and avoid unintended consequences and to deactivate or disengage misbehaving systems have clear implications for DT&E. The test strategy should include a means to characterize capabilities, limitations, and risks associated with these functions, including identifying means to:

- Identify and diagnose undesired behaviors.
- Characterize system potential for undesired behaviors.
- Evaluate the performance of human-AI teaming concepts.
- Characterize and assess means to disengage or deactivate systems when necessary.

For AWS, DoDD 3000.09 calls for very early engagement with DT&E activities. Before any decision to enter formal development of an AWS, the review panel established in accordance with DoDD 3000.09 must verify that:

- The system design incorporates the necessary capabilities to allow commanders and operators to exercise appropriate judgment over the use of force in planned applications.
- The system either completes engagements in time frames consistent with the commander’s intent, obtains further orders, or terminates the engagement.
- The system’s design and CONEMP account for risk to non-targets.

- Plans are in place for verification and validation (V&V) and T&E to establish system reliability, effectiveness, and suitability under realistic conditions (including possible adversarial action) sufficient to the potential consequences of unintended engagements or interference by unauthorized parties.
- This early review, and particularly the assessment of adequacy of test plans and V&V activities, requires substantive input from DT&E activities.

In addition, DoDD 3000.09 calls for a pre-fielding review to verify the adequacy and completeness of T&E and V&V activities. This pre-fielding review includes demonstration "that the system can be reprogrammed with sufficient rapidity to enable timely correction of any unintended system behaviors that may be observed or discovered during future systems operations," as stated in DoDD 3000.09. Although this is an operational test (OT) requirement, the entire history of DT&E findings over the course of system development will enable this requirement to be met more efficiently.

2.2.2.2 Robustness of ML-Enabled Systems

Although runtime assurance (RTA) can greatly improve system robustness, it introduces additional challenges for T&E—system mission performance now depends on a more complex system of systems. Diagnosis of the causes of system performance shortfalls and verification of worst-case system behavior may well be more difficult than for a system design that relies directly on an ML model, without runtime monitoring and intervention.

As discussed in Section 2.2.1, many ML models are prone to either high sensitivity to small changes in their inputs or worst-case errors that are large in magnitude and difficult to describe statistically. The term "robustness" is used in different technical communities to mean either insensitivity to small changes in input or worst-case performance that is still acceptably good. From a mission perspective, a system design is robust to the extent that it avoids unacceptable levels of performance across the entire mission context.

Many methods exist for quantifying the robustness of ML models in isolation. Most of these methods involve comparing model outputs for representative inputs against the outputs for slightly perturbed versions of those inputs. Model robustness testing can be informative not just for system characterization and risk assessment but also to inform engineering design choices and data collection needs (e.g., Wu 2020).

Assessment of system robustness is more challenging. The relationship between model robustness and system robustness may be complex, including the possibility that system component interactions and human-machine interactions (HMIs) may produce unintended and undesired emergent behaviors. In some cases, the worst-case model behaviors will occur in parts of the mission parameter space that are rarely experienced in actual operations and impose tolerable risk; in other cases, the model will misbehave intolerably, often at times critical to mission success. In such cases, a common design choice is to implement RTA—*runtime assurance*—in which an "online verification methodology [is used] to allow unproven autonomous controllers to perform within a predetermined envelope of acceptable behavior" (Gross et al. 2017; Schierman et al. 2020).

RTA seeks to mitigate the inherent brittleness of ML techniques by placing the unreliable (but high-performing on average) ML model in parallel with a less capable but more dependable model that can be used in situations where the preferred ML model is not trusted. Figure 2-1 illustrates the RTA architecture.

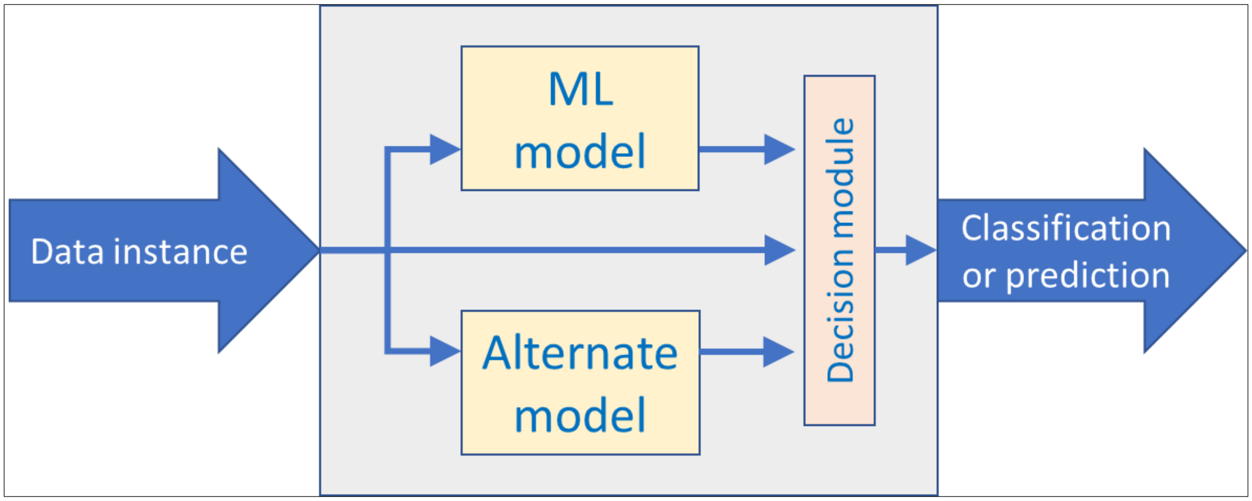


Figure 2-1. Runtime Assurance Architecture

The decision module (Seto et al. 1998) decides in real time whether the ML model is dependable for the current input and chooses whether to use the output from the ML model or the alternate model. The decision module might take into account the system state, the current input, the observed ML model output, metamodels or other XAI products, or some combination of those things. For example, the decision module might consider the following questions:

- Is the system in a state that was rare or unknown during learning?
- Is the input outside the envelope of training, validation, and test (TVT) instances used in learning?
- Is the output outside the envelope of outputs observed during learning?

- Is the ML output sensitive to very small input changes at this point?

Constructing a decision module able to answer these questions accurately may require the outputs of DT&E activities, such as those described in Section 3.5.

When RTA is feasible, it has obvious advantages for ensuring system robustness. The use of RTA, however, can complicate DT&E considerably, compared with the use of a single model. In addition to the original T&E challenge of characterizing ML model performance, RTA creates a system of systems with additional failure modes and potential emergent behaviors. Characterizing the resulting system capabilities, limitations, and risks may require additional instrumentation and more complex test designs, relative to a single-model architecture. In particular, DT&E will need to verify that:

- The monitor can reliably identify in real time when the ML model is and is not dependable.
- Switching from the ML model to the alternate model (and back) is timely, effective, and reliable. (This is particularly important if the output is being used for real-time control of a physical system.)
- The alternate model performs the mission adequately in those cases when it is invoked.
- No pernicious emergent behavior is exhibited in the interactions between the subsystems.
- If the alternate system is human operated, no human factors issues (e.g., attention, fatigue, response time, bias) exist in the intended CONEMP.
- The RTA system of systems can function effectively in the intended power and cooling context.

The use of RTA also has implications for DT&E supporting system safety, cyber, and airworthiness (where applicable) certifications. Testers should provide early feedback to system developers regarding the testability and cost or schedule implications of any planned use of RTA.

2.2.2.3 Technology Risk Characterization for AI-Enabled Systems

Technology risk assessments play an important role in defense acquisition, from early S&T research through prototyping and system development. Independent Technical Risk Assessments (ITRAs) (see Section 4272 of Title 10, United States Code (U.S.C.)) are required by statute for all programs and are conducted by the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) for all Acquisition Category (ACAT) ID programs. A Technology Readiness Assessment (TRA), often incorporated into the ITRA, is also mandatory if the program is a major defense acquisition program (MDAP). For applications where no non-AI alternative technologies can do the mission, the AI models constitute critical technology elements to be assessed as part of the TRA. These risk analysis activities can be informed by the outputs of DT&E, including test activities conducted during technology development and prototyping phases, before establishment of a program of record.

Assessing the degree of additional technology risk posed by the use of AI in a system design can be challenging, particularly for applications where the AI is embedded in a larger system (or system of systems). Once model development has begun, several predictable areas of technical risk can be explored through test. These include:

- **Performance**. The chosen ML techniques may not be capable of simultaneously achieving all mission goals for performance, safety, security, and suitability.
- **Human-machine interactions (HMI)**. The human-AI teaming concept should be an early design consideration. This consideration allows testing to determine the practicality and efficacy of use by humans under relevant conditions. If humans cannot interact with the model in the necessary and acceptable ways (given the expected knowledge and training of the human users), then redesign and rework will be required, even if the ML component exhibits the desired performance.
- **Brittleness**. If the model’s output can sometimes change sharply because of small changes in inputs, there is risk that overall system behavior will be too unpredictable for operational use. Testing will be needed to characterize if and when system responses are brittle and how those cases might affect operations. This testing might involve novel instrumentation of the ML model or the use of XAI techniques.
- **Generalizability**. For operational systems, the conditions under which they will be expected to perform well may be quite varied. Characterization of how model performance changes in different operational contexts and the potential impacts of those changes on mission performance will be an essential part of the DT&E mission to characterize system capabilities, limitations, and risks at both the model and system levels.

2.2.3 Informing Systems Engineering

During the system design process, DT&E can provide important information and feedback to the systems engineering process. In addition to general input into the testability of requirements, DT&E can contribute to:

- Determining technical feasibility for materiel solutions.

- Deriving ML model specifications that support mission goals.
- Informing VV&A of TVT data and ML models.
- Evaluating potential suitability issues in human-machine teaming (HMT) CONEMPs.

The following subsections focus on materiel solution technical feasibility; deriving AI model specifications from system requirements; VV&A of ML TVT data and models; and HMT.

2.2.3.1 Materiel Solution Technical Feasibility

DoDI 5000.02, “Operation of the Adaptive Acquisition Framework,” calls for assessment of the technical feasibility of proposed MDAPs; for systems planning to use AI, this may include assessment of whether available AI technology and data sources can support the intended mission. DT&E may be called on to support these assessments. Although confirmation that a materiel solution is technically feasible is required only for MDAPs, it is a best practice for all programs. Assessment of technical feasibility asks whether sufficient evidence exists to believe that existing technologies are adequate to achieve mission goals for a proposed system. Inadequate feasibility assessments have led to some of DoD’s most costly acquisition failures, in terms of dollars spent and time wasted.

For proposed systems intending to rely on ML technologies, three key feasibility assessments can be conducted very early, before analyses of alternatives (AoAs):

- Assessment of the sufficiency of available data to be used for TVT of the ML models.
- Comparison of the anticipated size, weight, power, cost and cooling (SWaP-C2) demands of the ML model against foreseeable platform constraints on those resources. The understanding of SWaP-C2 should be expanded to include computational resources—especially relevant to operation at the edge.
- Human factors evaluation of the intended CONEMP.

Training data sufficiency in this context includes quantity, quality, and relevance of the data to the mission requirements. This sufficiency may include the availability of raw data from the intended mission environment; data coverage (in terms of the range of capabilities the ML model is intended to provide); mission-relevant labels for SL; and credible methods or sources for synthetic data to supplement real-world data. Contractual rights to access TVT data may also affect both feasibility and test planning. VV&A of TVT data is addressed in more detail in Section 3.2.2.3.

SWaP-C constraints on deployed systems, especially systems intended for use at the tactical edge, also raise feasibility questions. Even before specific ML design choices, consideration can be given to the system or systems in which it will be embedded. Stable electrical power will be needed, as will access to computational power. Both of these can require cooling—especially within the confines of space-constrained tactical systems. Notional or prototype designs can be characterized in terms of the power and thermal management constraints that must be met as part of a feasibility assessment. DT&E can help to inform assessments of trade-offs between model capability and available SWAP-C using current and foreseeable technologies and to assess whether current capabilities are reasonably likely to be able to support the intended missions within the required form factors.

Human-AI teaming also introduces novel, specific challenges in human-factors engineering. Allocation of tasks between humans and AIES must be accomplished in a way that does not overly tax the abilities of humans or systems and that conforms with RAI goals relating to governability of AIES and maintaining effective human command and control. For systems where operator trust in the system affects operational effectiveness, the CONEMP must be carefully designed to support proper calibration. Feasibility assessments of proposed CONEMPs will often be possible even before any notional ML design or architecture has been selected.

Once the initial model development has begun, T&E to assess HMI and brittleness will be essential to any feasibility assessment. Applications with significant HMT require particular attention with respect to feasibility (see Section 2.2.3.4).

2.2.3.2 Deriving AI Model Specifications from System Requirements

Substantial academic and commercial literature exists on the performance evaluation of ML models. Much of this literature focuses on model-level performance—i.e., how well the trained model performs at the task it was trained to do, as measured by its performance on a set-aside validation dataset (during training) or test dataset (after training). Note that "test dataset" here refers to data that will drive the AI model, not the performance data that can be collected once the model is exercised. Numerous measures of performance have been proposed for both supervised and unsupervised ML.[1] Section 3.2.4 discusses several of the more common performance metrics in use.

Although it is useful for developmental testers to understand ML model metrics, it is even more important for testers to assess whether the metrics being used by the program office and the development team are correctly aligned with mission requirements. A common failure mode in the use of ML is for model performance specifications (or contractor incentives) to be set in

[1] Reinforcement learning (RL) performance measures are specific to the mission they are intended to enable; no generic performance metrics exist for RL beyond "success rate."

ways that do not support effectiveness and suitability in mission contexts or are inconsistent with the intended CONEMP (Ryseff et al. 2024).

As an example, consider a hypothetical chemical agent detection system that collects periodic samples from the air and analyzes them using ML. If model specifications or contractor incentives are based on achieving a high probability of successful detection in the event of a chemical attack, the training process will highly penalize missed attacks. This penalization will tend to increase the false alarm rate for the system—possibly to the point where it becomes unusable for actual missions. Even if developers are using an objective function that incorporates both probability of detection (P_d) and false alarm rate, the relative weights given to those outcomes might not reflect commanders' actual operational preferences.

Testers need to be aware of what measures the developers are trying to maximize and what incentives the contractors have been given. PMs should be alerted to potential mismatches. It will almost never be operationally ideal to train a model to maximize a generic balanced metric even using standard developer metrics such as F_1 score or cross-entropy loss—much less a single metric like P_d or accuracy (Mao et al. 2023). It may be useful for developers to deliberately train a variety of nonoperational models using different penalties on the various error modes to map out the receiver-operating characteristic (ROC) curve of possible performance (e.g., Sokolova et al. 2006). It may be that the knee in the curve, even though it seems to provide the best trade-off between error modes, would not be the most operationally effective or suitable model for the intended mission.

Similarly, for ML models that predict numerical values, standard regression measures such as mean squared error (MSE) might not be the most appropriate. As noted in Section 2.2.2.2, ML models often have unusual error distributions and may produce occasional outputs that are wildly wrong. Robust optimization techniques exist that allow developers to accept some amount of increased average error to avoid these extreme errors (e.g., Sun et al. 2020).

It is sometimes possible—for systems where the limits of acceptable error are known—to tune the training to maximize average performance subject to ensuring acceptable performance. Identifying the acceptable limits of various potential errors in various mission contexts is not, in general, something the ML developers can do for themselves. DT&E, possibly involving modeling and simulation (M&S), can be invaluable in aligning model specifications with mission priorities.

2.2.3.3 VV&A of ML Training, Validation, and Test Data and Models

DoDI 5000.61, “DoD Modeling and Simulation Verification, Validation, and Accreditation,” states that models, simulations, distributed simulations, and associated data used to support DoD processes, products, and decisions:

- Undergo V&V throughout their life cycles.
- Are accredited for a specific intended use.

For the specific case of AI models, this policy requires VV&A of the data used to develop and assess the trained model, as well as VV&A of the model itself. Although VV&A of simulation models used in T&E is a familiar process, VV&A of ML models and their associated data is new. DT&E may be called upon to support the VV&A processes, and DT&E outputs may be needed to validate and accredit ML data and models.

Data verification asks this question: Are these data suitable for training, validating, and testing an ML model? In particular, are the data:

- Correct (and correctly labeled)?
- Appropriately formatted?
- Appropriately normalized and scaled?
- Relevant?
- Sufficient (in quantity and variety)?
- Free from data poisoning?
- Free from unwanted bias?

Data validation goes beyond mere correctness to ask the following question: Are these data mission-appropriate for the intended mission and environment? That is:

- Do they fully cover the intended mission space?
- Are they statistically representative with respect to relevant labels or events?
- Are all mission-important features labeled?
- Are any synthetic data operationally equivalent to real data?
- Have classification, privacy, and proprietary data concerns been mitigated?

Table 2-2 describes V&V roles for key data features.

Table 2-2. Verification and Validation Roles for Key Data Features

Data	Verification	Validation
Labeling	Accurate	Mission relevant
Formatting	Practical for evaluating the data and training a model	N/A
Quantity	Sufficient to train a model with the desired functions	Comprehensive of the intended mission space
Content	Consistent with current AI standards	Relevant to intended mission space
Statistics	Free from poisoning and unwanted bias	Statistically representative and operationally equivalent to real mission inputs
Lawfulness	N/A	Mitigate classification, proprietary, and privacy concerns

A best practice for ML development is to provide *data cards* that document TVT data sources and data preparation steps used in the development of each ML model.[2] These data cards can inform test planning by highlighting potential data-related issues or risks.

A data card should ideally include:

- The source of the data, ideally with contact information.
- What purpose the data were collected for.
- When the data were collected.
- A description of the data instances and labels, including examples.
- A description of all data transformations, normalization, and augmentation (including any synthetic data) used to develop the dataset.
- Any known limitations or risks associated with the dataset (including privacy or security concerns).

DT&E activities support both verification and validation. In particular, DT&E can review any data cards provided by the developers and can conduct exploratory data analysis to help identify potential issues with incorrect or incomplete data; data coverage and representativeness issues; and unwanted bias. Detecting and mitigating bias is a particularly challenging task, especially in cases where a possibility of bias exists against groups that are not identified explicitly in the TVT

[2] See, for example, https://sites.research.google/datacardsplaybook/.

data. The Chief Digital and Artificial Intelligence Office (CDAO) provides links to several tools for detecting and mitigating bias as part of its RAI Toolkit.[3]

Model verification establishes that learning has been achieved—the model is able to perform its intended task at a reasonable level against validation and test sets that were not used in training. Appropriate performance measures for this assessment are discussed in Section 3.2.4.

Similar to the data cards described earlier in this section, a best practice is for developers to provide *model cards* that describe what a model is; what it is supposed to do; and any known shortcomings or issues at a high level (Mitchell et al. 2019). Model cards can be useful in tracing model performance from requirements to output, in support of the traceability goal of the DoD AI Ethical Principles, as outlined in the May 26, 2021, DepSecDef Memorandum. The existence of a detailed model card is a positive indicator for contractor performance and process maturity.[4]

Model validation establishes not only that the ML model is well-documented and performing well against its training objectives but also that the model performance is adequate for the intended operational environment and CONEMP. The challenges associated with aligning model performance and mission needs are discussed in Section 2.2.3.2. DT&E can play an important role in providing early feedback to developers regarding any potential mismatch of training objectives and mission needs and helping to establish minimum acceptable levels of performance for various model-level metrics, given mission-level goals.

Verification and validation are fundamentally evaluation activities, naturally incorporating the outputs of DT&E. Ideally, DT&E personnel should be involved in V&V planning, to help generate the needed information efficiently in conjunction with other T&E activities. Accreditation, in contrast, is an approval process with authorities determined by the various DoD Components. DT&E personnel may or may not be designated as approval authorities for data and ML models; regardless, V&V outputs and analyses, in conjunction with other evaluations and assessments such as TRAs and ITRAs, should support the accreditation by any accrediting authority.

2.2.3.4 Human-Machine Teaming

HMT is when a human works with an AIES with interactions beyond established practices for use of a machine as a tool. Assessments of the teammates—the human and the machine—are not changed much by this interaction. The ability to assess the quality of a team is new and remains incomplete, but it is a research area. Metrics are in development to address perspective,

[3] https://rai.tradewindai.com/.

[4] For additional information on model and data cards, see https://www.kaggle.com/code/var0101/model-cards.

cooperation, and resources within the team. These points are discussed below, with additional references provided at the end of this subsection.

Contemporary AI systems give rise to qualitatively different HMT compared with when machines were simply treated as tools. Machines now have some degree of agency, allowing them to act on their own. Testing of the human-machine system will, therefore, require an understanding of the characteristics of the team members and observation of team decisions and actions as well as the interaction between humans and machines. This interaction between human and machine as teammates is fundamentally new for the test community—for example, when responsibilities are shared and changing or when decisions requiring collaborative actions must take place under conditions of degraded communications. The resultant need to characterize the effectiveness of this interaction has significant implications for T&E support to systems engineering.

Some of the new complexities that arise in HMT interactions derive from the differences in machine and human cognition. For example, machine cognition is generally algorithmic and fixed, whereas humans can adapt their approaches based on experience and judgment. (Machines that actively learn would blur this distinction, but that is beyond the scope of this guidebook.) These complexities make it difficult to anticipate or diagnose failures in HMT. The interactions between ML-enabled machines and humans, coupled with the fragility of ML systems as well as human variability, make evaluations based solely on observing test outcomes insufficient.

Because of differences in human and machine cognition, machine errors are challenging for humans to anticipate or diagnose, affecting both development and employment. XAI may help mitigate this issue, but it depends on timely and accurate communications between machine and human. Work in this area has largely focused on explanation to users, with limited applicability to test planning or diagnosis.

The assessment of individual behavior of humans and machines within an HMT does not radically differ from “ordinary” assessments. The HMT framing, however, helps emphasize that the team interactions, between individuals, are key. This framing is where unfamiliar metrics are needed. Early in development, it is essential for iteration on design, test, and CONEMP development. Later in a program, it is essential for the critical operational issues (COIs) to be assessed over a relevant range of conditions.

Metrics that address the quality of HMT are essential for effective testing. Although separate metrics for human and machine performance are routinely included in T&E, the metrics for HMT are still evolving. For humans, these include metrics on trust, trust calibration, automation bias, and algorithm aversion bias. In addition, established metrics also exist for human stress and workload, both of which can impact HMT functioning (e.g., Damacharla et al. 2018).

The presence of AI introduces new elements to the interaction among the teammates. Characterization of the team interaction element of HMT remains incomplete, especially with respect to metrics for T&E. Candidate high-level metrics for team interaction should include how the team members view their circumstances; how they work together; and how they share tasks within the team. These elements can be captured as:

- Team perspective.
- Cooperative behavior.
- Resourcingallocation within the team.

Team Perspective

For HMTs, the "team perspective" is the union of the perspectives of the team members. It depends upon situational awareness (SA) and information accuracy, both of which have established metrics that can be determined in testing. This unified perspective has the potential for significant differences and inconsistencies between the team members, which is nevertheless the basis for team decision making and action. Fundamental differences exist between the processes that underlie human cognition and those that underlie machine cognition, from their material substrate to how they perceive and determine relevance. The resulting cognitive differences in humans and machines will preclude a completely shared SA. Similarly, the information available and capacity for information storage, as well as information accuracy, may differ between teammates. One metric that may capture this is "information coherency," which has been defined as "the degree to which information needed for one agent's work is readily available to that agent and remains predictable and accessible to them" (Ma et al. 2022).

Despite these differences in available information and incompletely shared SA, the team must act to support the mission. Informatiom accuracy can be readily assessed *for the test conditions* by looking at outcomes. The ability to generalize to other test conditions depends upon measurements determining the quality of the SA and the accuracy of the information available to the team members who need it.

Outcomes of testing alone will be insufficient. The coverage is too sparse. A good outcome, or even a good decision leading to a desirable outcome, has limited applicability with an AIES. It is also necessary to determine whether the decisions were made for valid reasons. Decision-making algorithms will need to be instrumented to provide insight into the quality of those algorithms. In general, these decisions will require insight into the SA and information accuracy.

In early development, instrumenting the machine decision making to include the effects of SA and information accuracy on that decision making is a testing challenge. This instrumentation should provide insight into what influences SA and information sharing, allowing for generalization of performance beyond the test conditions.

Cooperative Behavior

Cooperative behavior depends on a team member's ability to shift an activity—transferring responsibility and control between human and machines. An example might be an AI-managed sensor teamed with a human surveilling an extensive area. Is the area of regard shared or split? Static or evolving? If evolving, which teammate determines the evolution? The timing and method of this transfer will influence team performance. Team performance relies on the mechanism teammates use to affect each other's actions and on the extent of their influence on each other. The team must remain cohesive, united in its objectives. Differences in cognition, such as machine rigidity versus human flexibility, can undermine cohesion. In the surveillance example, hese differences might be especially problematic in cases where there are shared or changing areas of regard for the machine and the human. The combination of agency shifting and team cohesion is crucial for cooperative behavior. In testing any teaming scenario, it is important to assess not only mission outcomes but also teaming factors such as cohesion. Thus, having metrics for these aspects is critical for the execution of the iterative design cycle to improve teaming.

Resourcing

Enabling an effective team perspective and effective cooperation depends upon adequate resourcing of the teammates. Strictly speaking, the metrics here are specific to the team members, rather than to the interaction. Resources must be sufficient to continue tasks under stress. For humans, the effects of the physical environment, cognitive load, and other stressors are well studied, with established metrics. Machines also need resources such as power, processing capability, and temperature control. T&E can explore and document these features with established techniques.

Effective resource support of the humans and machines will enable the team to deal with deviations from optimal conditions, where the team must determine how to allocate resources. One such resource unique to teaming is the rarely considered resource of attention. Therefore, joint, collective attention, rather than individual team member attention, must be prioritized and used on the most important tasks. Instrumentation to measure where attention is focused will be essential.

HMT Research and References

Although these three areas—team perspective, cooperative behavior, and resourcing—have been identified and some metrics are already deployed, designing and assessing HMTs incorporating contemporary AI capabilities is still an active area of research. Much of this research builds on human-robot interactions and human-human teams. The areas of “Team Perspective” and “Cooperative Behavior” already have extensive work on metrics to assess these features. However, the link between the resources of the teammates and the effectiveness of the team remains relatively unstudied. Understanding this link is important for any evaluation that requires generalizing the results of specific scenario testing to a wider set of conditions. Furthermore, until recently, roles in HMT operations were essentially fixed, but shifting responsibilities is becoming increasingly relevant and will need to be addressed.

Again, these are active research areas. The following documents provide additional information on metrics relevant to HMT:

- The Systems Engineering journal article, “A Team-Centric Metric Framework for Testing and Evaluation of Human-Machine Teams” (Wilkins et al. 2024).
- The National Academies of Sciences, Engineering, and Medicine book, *Human-AI Teaming: State-of-the-Art and Research Needs*.
- The Human-Intelligent Systems Integration journal article, “Human-Machine Teaming: Evaluating Dimensions Using Narratives” (Lyons and Wynne 2021).

2.2.4 Informing Acquisition Management

DT&E also provides valuable input to PMs for a variety of acquisition decisions, ideally starting before establishment of a formal program of record. The set of acquisition decisions and reviews that should be informed by developmental test (DT) outputs includes:

- AoAs.
- Milestone decisions.
- Operational Test Readiness Review (OTRR).
- Contract structures and data rights.
- Schedule estimation.

This following subsections consider DT roles related to each of these acquisition decisions and reviews.

2.2.4.1 Analysis of Alternatives

AoAs are conducted for major programs to explore a trade-space of cost, schedule, and performance among options for providing needed capabilities. The T&E community can help assess the unique and distinguishing aspects of the technology and other risks of using AIES that most influence those trade-offs and options. Risks evaluated should include those associated with obtaining, preparing, and conducting VV&A of datasets; conducting VV&A of ML models; and comprehensively evaluating the performance of the overall system using embedded AI.

AoAs are conducted for MDAPs. The most recent versions of DoD issuances describe the conduct of an AoA.[5]

- DoDI 5000.84, "Analysis of Alternatives," states the following:
  - At minimum, the AoA study guidance will require a trade-space analysis of cost, schedule, and performance. The performance analysis will include both the capability of the options examined and the effect each option has on mission accomplishment.
  - The DoD Components should ensure that the AoA study team conducts sensitivity analysis to identify the dependence of results on key parameters of the analysis. This enables decision makers to understand whether the solutions examined are robust.
  - The AoA study team should consider cost and affordability as early in the analysis as possible.
  - The Director of Cost Assessment and Program Evaluation (DCAPE) will ensure that the AoA study guidance establishes a study advisory group to oversee the conduct of the AoA.
- The DoD issuances that establish and assign associated Under Secretary of Defense for Research and Engineering (USD(R&E)) responsibilities and activities state that the USD(R&E):
  - Conducts assessments of technology, engineering, integration, and manufacturing risks, in accordance with DoDD 5137.02, "Under Secretary of Defense for Research and Engineering (USD(R&E))," and consistent with the ITRA requirements in Section 4272 of Title 10, U.S.C.

[5] Previous guidance in the December 2008 revision of DoDI 5000.02 that the AoA "assess the critical technology elements associated with each proposed materiel solution, including technology maturity, integration risk, manufacturing feasibility, and, where necessary, technology maturation and demonstration needs" is no longer provided in current DoD directives or instructions.

  - Advises the DCAPE in the USD(R&E) areas of responsibility in preparation of the AoA study guidance, in accordance with DoDI 5000.84.
  - Confirms that a materiel solution that addresses the validated need or capability gap for the MDAP is technically feasible and achievable, in accordance with DoDI 5000.02.
- The forthcoming DoDI 5000.DT states that the DT&E program will provide information for cost, performance, and schedule trade-offs.

The alternatives considered in the AoA may differ in how they are expected to use AI capabilities, or even whether they use AI at all. Specific features of proposed AIES may drive cost and schedule in both development and testing activities. These cost and schedule impacts may arise in data preparation and other early activities; M&S validation and accreditation; ML model assessment; or late stage assessment of complete systems or major components.

Consistent with the above, as well as with the other sections of this guidebook, the Office of the Director for Developmental Test, Evaluation, and Assessments (DTE&A) should consider conducting the following activities supporting USD(R&E) advice to the DCAPE on AoA study guidance dealing with an AIES or MLES:

- Prepare assessments of issues regarding the sufficiency—as well as the feasibility and schedule implications of conducting VV&A—of the datasets needed to develop the AIES or MLES alternatives prescribed in the study guidance consistent with DoDD 3000.09 and DoDI 5000.61.
- Prepare assessments of issues regarding the sufficiency of VV&A of any simulation models expected to be used to either train or evaluate the AIES or MLES alternatives, including M&S previously accredited for testing of non-AIES or MLES, in accordance with DoDI 5000.61.
- Prepare assessments of the feasibility—as well as the schedule and test infrastructure implications of conducting VV&A—of the ML models associated with the AIES or MLES alternatives prescribed in the study guidance consistent with DoDD 3000.09.
- Provide assessments of the feasibility—as well as the schedule and test infrastructure implications—of conducting overall system- and subsystem-level T&E of the AIES or MLES alternatives prescribed in the study guidance consistent with DoDD 3000.09.

In each case, these assessments should identify the unique and distinguishing aspects of the technology and other risks associated with the AI/ML implementations to be used for the alternatives that most influence cost and schedule trade-offs. The assessments should be updated as appropriate to support USD(R&E) participation in the AoA study advisory group as the analysis proceeds. Final versions of the assessments should be prepared for use by the USD(R&E) and potential submission to DCAPEfor assessing the adequacy of the completed AoA. The final versions of these assessments can also be used, as appropriate, to support the approvals and certifications required by the USD(R&E) before formal development of the AIES or /MLES begins (i.e., Materiel Development Decision (MDD) approval, in accordance with DoDD 3000.09).

2.2.4.2 Milestone Decisions

Milestones for major capability acquisition (MCA) programs (ACAT I, ACAT II, and automated information systems not proceeding using other acquisition pathways) and the information needed to support the associated decision making include the major reviews discussed in this section which are conducted in accordance with DoDI 5000.85, “Major Capability Acquisition.”

The MDD is the nominal entry point into the acquisition process for MCA programs. It is informed by an Initial Capabilities Document; AoA study guidance (upon which the USD(R&E) advises; see Section 2.2.4.1); and an AoA study plan.

The MDD nominally precedes initiation of the Materiel Solution Analysis phase, during which the AoA and other analyses are conducted focusing on key trades between cost and capability, life-cycle cost, schedule, concepts of operations, and overall risk.. Other analysis conducted include the following: Affordability analysis; early systems engineering analysis; threat projections; product support and sustainment planning; independent cost estimate (ICE) (see Section 3222 of Title 10, U.S.C.); and ITRAs.

Milestone A approves program entry into initiation of the Technology Maturation and Risk Reduction (TMRR) phase, the program acquisition strategy, and release of the request for proposals (RFP) for conducting TMRR activities; the latter two are informed by an approved draft Capability Development Document. RFPs for AIES can inadvertently disclose to the public (and our adversaries) the current gaps in AIES capabilities and identify the DoD's warfighting requirements for specific and sensitive activities. Acquisition teams should work with their covering classification offices to ensure RFPs aren't inadvertently creating identifiable harm to national security and should consider a layered approach of general requirements to public sources, closed (registration-required/invite-only) industry days, and classified RFPs when required to protect national security

Considerations at Milestone A include:

- The affordability and feasibility of the preferred alternative derived from the results of the AoA.

- The technologies that must be matured during TMRR phase.
- The remaining trade-space for achieving the needed military capability and priorities within the trade-space.
- The remaining technical, cost, and schedule risks and the plans (and associated resources) to address them.
    - Determination with a high degree of confidence that the technology to be developed will not delay the program's fielding or that any technology that could cause a delay will be matured and demonstrated in a relevant environment separate from the program using the appropriate authorities (see Section 4251 of Title 10, U.S.C.).
- The acquisition strategy, including plans for dealing with intellectual property (IP) issues.
- The test strategy.
- The life-cycle mission data plan.

During the TMRR phase, work focuses on reducing technology, engineering, integration, and life-cycle cost risk to the point that a decision to contract for Engineering and Manufacturing Development (EMD) can be made with confidence in successful program execution for development, production, and sustainment. Another ICE and ITRA are conducted and a test and evaluation master plan (TEMP) is prepared.

The decision to release a development (i.e., EMD) RFP is the point at which the proposed RFP is reviewed to ensure that its content comprises an executable and affordable program using sound business and technical approaches, and that work accomplished during TMRR has been sufficient to support successful development. Criteria for Milestone C (low-rate initial production) approval including T&E results and funding requirements are documented as part of the decision.

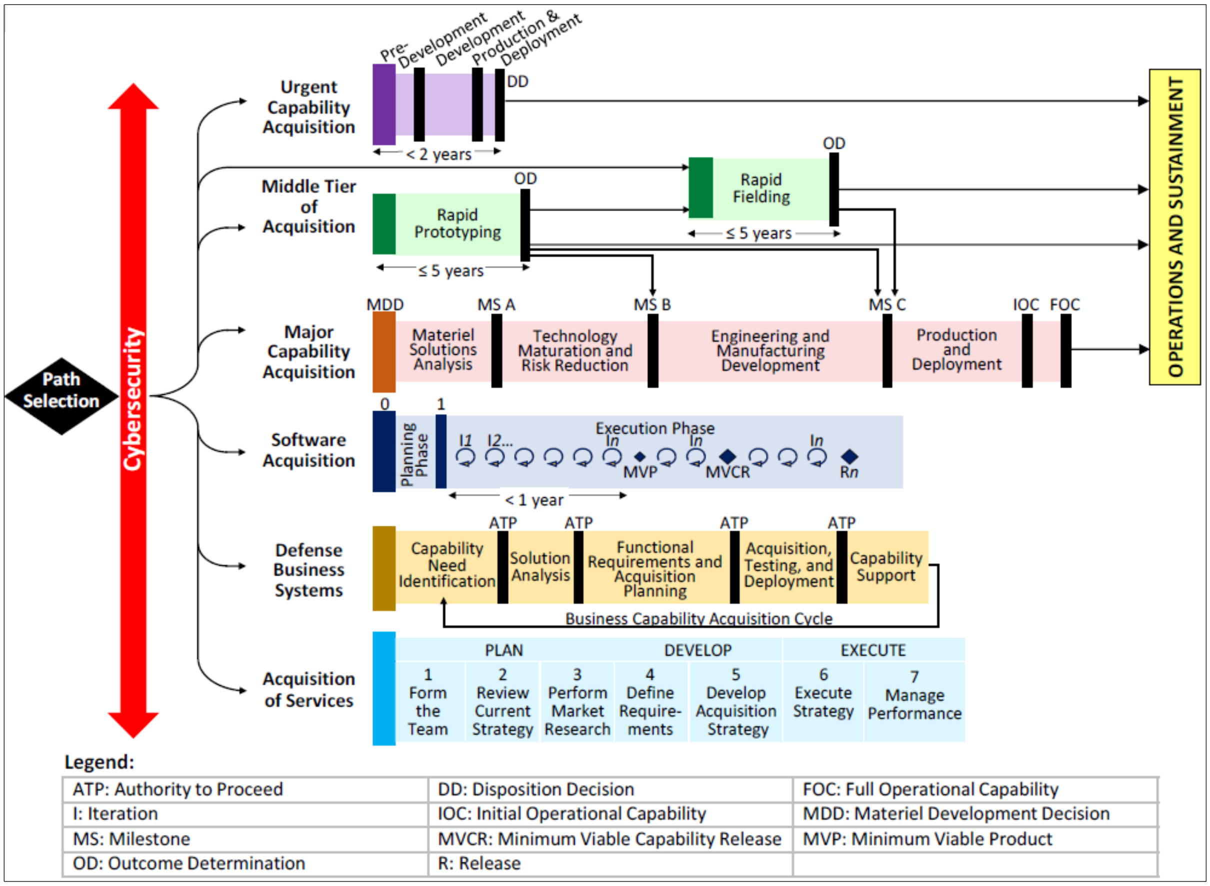
Milestone B approves entry into the EMD phase. Requirements for approval may have been satisfied at the RFP release decision point:

- Requirements include demonstration that all sources of risk have been adequately mitigated to support a commitment to design, development, and production.
- Risk sources include, but are not limited to, technology, threat projections, security, engineering, integration, manufacturing, sustainment, and cost risk.

During the EMD phase, detailed design of all hardware and software is accomplished; the technical baseline is established; pre-production hardware is fabricated; and software is coded. DT&E provides hardware and software feedback to the PM on the progress of the design process and on the product’s compliance with contractual requirements, effective combat capability, and the ability to achieve key performance parameters (KPPs) and key system attributes (KSAs). Criteria for ending EMD include a stable design that meets requirements as demonstrated by the results of T&E; manufacturing processes under control; effective software sustainment processes; and available industrial production processes. Milestone C approves initiation of low-rate initial production (or limited deployment for an automated information system) and entrance into the production and deployment phase. Considerations include the results of DT and early operational assessment; software maturity; and any remaining risks.

The full-rate production (FRP) decision reviews the results of operational test and evaluation (OT&E) and progress in manufacturing to decide whether to produce at full rate. Proceeding to FRP requires control of the manufacturing process; acceptable performance and reliability; the establishment of adequate sustainment and support systems; and for MDAPs, an ICE and an ITRA. The Operations and Support phase follows FRP.

Figure 2-2 includes other decision points as well. In addition to the MCA pathway, the Adaptive Acquisition Framework includes urgent capability acquisition (see DoDI 5000.81, “Urgent Capability Acquisition”); middle tier of acquisition (MTA) rapid prototyping andfielding (see DoDI 5000.80, “Operation of the Middle Tier of Acquisition”); software acquisition (see DoDI 5000.87, “Operation of the Software Acquisition Pathway”); defense business systems (DBS) (see DoDI 5000.75, “Business Systems Requirements and Acquisition”); and acquisition of services (see DoDI 5000.74, “Defense Acquisition of Services”). The first three of these pathways incorporate outcome decisions that can lead to entry into the MCA pathway or directly to fielding. Software acquisition incorporates decisions regarding minimum viable products, minimum viable capability releases, and releases; and DBS incorporates decisions regarding authority to proceed (ATP).


Path Selection
Cybersecurity
Urgent Capability Acquisition
Pre-Development
Development
Production & Deployment
DD
< 2 years
Middle Tier of Acquisition
Rapid Prototyping
OD
≤ 5 years
Rapid Fielding
OD
≤ 5 years
Major Capability Acquisition
MDD
MS A
MS B
MS C
IOC
FOC
Materiel Solutions Analysis
Technology Maturation and Risk Reduction
Engineering and Manufacturing Development
Production and Deployment
Software Acquisition
0
1
Planning Phase
Execution Phase
I1
I2...
In
In
In
MVP
MVCR
Rn
< 1 year
Defense Business Systems
ATP
ATP
ATP
ATP
Capability Need Identification
Solution Analysis
Functional Requirements and Acquisition Planning
Acquisition, Testing, and Deployment
Capability Support
Business Capability Acquisition Cycle
Acquisition of Services
PLAN
DEVELOP
EXECUTE
1 Form the Team
2 Review Current Strategy
3 Perform Market Research
4 Define Require-ments
5 Develop Acquisition Strategy
6 Execute Strategy
7 Manage Performance
OPERATIONS AND SUSTAINMENT
Legend:

ATP: Authority to Proceed	DD: Disposition Decision	FOC: Full Operational Capability
I: Iteration	IOC: Initial Operational Capability	MDD: Materiel Development Decision
MS: Milestone	MVCR: Minimum Viable Capability Release	MVP: Minimum Viable Product
OD: Outcome Determination	R: Release	

Source: DoDI 5000.85

Figure 2-2. Adaptive Acquisition Framework

In accordance with DoDD 5137.02, USD(R&E) authorities and responsibilities related to milestone decisions for ACAT ID MCA programs include the following:

- Advises the Under Secretary of Defense for Acquisition and Sustainment (USD(A&S)) regarding MDDs, milestone decisions, production decisions, and technology insertion strategies on ACAT ID programs.
  - Conducts assessments of technology, engineering, integration, and manufacturing risks.
  - For Milestone B and C, provides an assessment of the sufficiency of DT&E plans, including the use of automated data analytics or M&S tools and methodologies.
  - Participates in MDAPs Configuration Steering Boards and Defense Acquisition Boards throughout the program life cycle to advise on technical risk and impacts to cost, schedule, and performance.

  - Prepares the post-Preliminary Design Review assessment before Milestone B to support the USD(A&S) determinations, in accordance with Section 4252 of Title 10, U.S.C.
- Conducts and approves ITRAs for ACAT ID programs.

In accordance with DoDD 5137.02, other USD(R&E) responsibilities and authorities potentially related to any of the decision points displayed in Figure 2-2 for any pathway include the following:

- In coordination with the Under Secretary of Defense for Intelligence and Security and other senior officials, where appropriate, supports MDAPs (see Section 2430 of Title 10 U.S.C.) and other acquisition programs (i.e., any program using any acquisition pathway) in the areas within which the USD(R&E) has direct or shared mission equities, including:
  - Assured software and hardware.
  - Program and technology protection.
  - Technical information and data protection.
  - Technology vulnerability and exploitation mitigation and assurance.
  - Anti-tamper, reliability, maintainability, and supply chain risk management.
  - Cybersecurity and resiliency.
  - Program protection planning.
  - Conduct of program and technology assessments, systems engineering, technical risk, joint mission engineering, joint architectures, prototyping and experimentation outcomes, and technology-related recommendations.

In addition to the responsibilities and authorities listed above, DoDD 3000.09 assigns the USD(R&E) other AI/ML-specific responsibilities including developing testable requirements for AIES or MLES to ensure that those systems are implementing the DoD AI Ethical Principles and the DoD Responsible AI Strategy and Implementation Pathway [6]. In conjunction with the Under Secretary of Defense for Policy (USD(P)) and the Vice Chairman of the Joint Chiefs of Staff (VCJCS), before MDD, the USD(R&E) must also review and approve the plans for developing, as well as testing, AIES or MLES to ensure that those plans comply with the requirements in DoDD 3000.09. AIES or MLES programs must be reviewed again by the USD(A&S), VCJCS, and USD(P) for compliance with DoDD 3000.09 before fielding. Given USD(R&E)

[6] The T&E implications of the DoD AI Ethical Principles, the DoD Responsible AI Strategy and Implementation Pathway, and other AI/ML-specific requirements are discussed in greater detail in Sections 2.2.1, 2.2.2, and 2.3 of this guidebook.

responsibilities for monitoring AIES or MLES to identify when changes to the system may require additional T&E (see Paragraph 2.3.h. of DoDD 3000.09), it is likely that the USD(R&E) will provide input to that pre-fielding review.

Implications of Milestones and USD(R&E) Responsibilities for DTE&A AI/ML-Related Activities

To support milestone decisions on the progress of programs using AI, the T&E community should assess the sufficiency of test planning and results to provide evidence that the program is adhering to the DoD AI Ethical Principles throughout the acquisition life cycle; assess the sufficiency of the data used for TVT of ML models, including the data's VV&A; evaluate the robustness and brittleness of trained ML models, their VV&A, and their readiness for integration within a system; and evaluate the performance of the integrated system with embedded AI, including the risks of unintended or harmful behavior.

The USD(R&E) has specific responsibilities for supporting milestone decision making on ACAT ID programs as well as a broader mandate for supporting any of the decisions depicted in Figure 2-2 associated with other acquisition programs in the areas within which the USD(R&E) has direct or shared mission equities, in accordance with DoDD 5137.02. These supporting activities are for all programs, including those incorporating AI/ML. Given the requirements of DoDD 3000.09 for ensuring AIES or MLES adherence to the DoD AI Ethical Principles and the DoD Responsible AI Strategy and Implementation Pathway, evidence of that adherence and the associated risks will be topics of interest at all of the milestones and decision points displayed in Figure 2-2, as well as at the pre-MDD and pre-fielding reviews mandated by DoDD 3000.09.

Regarding AIES or MLES and DTE&A efforts in particular, assessments conducted to support decision making can include:

- A pre-MDD review of compliance with DoDD 3000.09 requirements, including adherence to the DoD AI Ethical Principles and the DoD Responsible AI Strategy and Implementation Pathway for developing the AIES or MLES, to assess the treatment of:
    - Datasets to be used.
    - AI models to be implemented.
    - Evidence to be generated by T&E.
    - Conduct of VV&A.
- At MDD, or the earliest decision point for other non-MCA programs:
    - Feasibility of alternative approaches for implementing AI/ML to be considered in the AoA for MDAPs or the development approach to be used in other programs.

  - Availability and sufficiency of the associated datasets for training and evaluation of AI/ML models.
  - Alternative approaches and potential issues associated with conducting VV&A of the associated AI/ML datasets and models (see Section 2.2.3.3).
  - Potential issues and risks associated with achieving and demonstrating the autonomous system attributes and fulfilling the other requirements specified in DoDD 3000.09 (also see Section 2.3).

- At Milestone A, Milestone B/EMD RFP Release, and Milestone C, as well as at outcome decisions in MTA, release points in software acquisition, and ATPs in DBS:
  - Adequacy of plans, physical infrastructure, and M&S for conducting DT&E of the AIES/MLES, including VV&A of the associated models and datasets.
  - Progress in conducting the VV&A, including issues that have arisen and any associated risks (see Section 2.2.2).
  - Implications of the T&E results for characterizing performance of the AIES/MLES and remaining risks in completing its successful development (see Section 2.2.1 and Section 2.2.2).
  - Risks remaining in the ability to conduct the T&E needed to demonstrate required performance, including the attributes and other AI/ML-specific requirements specified in DoDD 3000.09.
- At FRP or the latest decision points for non-MCA programs:
  - Implications of the T&E results for meeting all performance requirements and system attributes.
  - Any remaining risks to maintain required performance over the system's lifetime, including evolution of threats, including cyber threats.
  - Adequacy of plans for revising or maintaining and conducting T&E of AI/ML datasets, models, and the overall system, particularly if the AIES/MLES will continue to learn. Compliance with the AI/ML-specific requirements of DoDD 3000.09 will again be a topic of interest.
- Before fielding, another review for compliance with DoDD 3000.09, including evidence provided through T&E of that compliance, as well as the sufficiency of plans for continued monitoring of the system's behavior to determine when changes warrant additional T&E to ensure continued compliance.

Although the pre-MDD and pre-fielding reviews are mandatory for all AIES/MLES programs, not all the assessments described above will be needed at every decision point displayed in

Figure 2-2 for every program incorporating AI/ML, particularly if the programs are not ACAT I. Nonetheless, given the broad mandate that the USD(R&E) has for providing support to senior decision making, as well as the special requirements DoD has levied upon the attributes of AIES/MLES (see DoDD 3000.09), DTE&A and the Service test organizations and program offices should continually collaborate and be prepared to provide the assessments indicated above if requested by the USD(R&E) or other senior acquisition officials regardless of the pathway and ACAT of the program. In particular, regardless of the acquisition pathway that an AIES/MLES program is using, decision makers will have continued interest in the evidence provided by T&E of continued compliance with the DoD AI Ethical Principles, the DoD Responsible AI Strategy and Implementation Pathway, and other requirements contained in DoDD 3000.09.

As defined in forthcoming DoDI 5000.DT, the Integrated Decision Support Key (IDSK) is a table that identifies DT, OT, and live-fire data requirements needed to inform critical acquisition and engineering decisions (e.g., milestone decisions, key integration points, and technical readiness decisions). The IDSK is developed by the PM in consultation with the chief developmental tester, the chief engineer, and the operational test agency representative, in accordance with forthcoming DoDI 5000.DT. The Defense Acquisition University webinar, "Test and Evaluation as a Continuum: The Integrated Decision Support Key (IDSK) for Test Planning," and DoD issuances and guidebooks for DT provide additional discussion of the IDSK in the context of DT&E as a continuum.[7]

An IDSK should be produced early for any program and updated continually as the program progresses, design changes occur, and information accrues from development activities including T&E; AIES/MLES programs are no exception. The information needed to support the approvals and certifications mandated in DoDD 3000.09, in addition to information otherwise needed for any program and described above, will place substantial demands on T&E (and the associated resources) early in and throughout AIES or MLES programs. Reflecting those demands for information as specifically as possible in an up-to-date IDSK is particularly important for conducting realistic schedule and resource planning for such programs. The IDSK, continually updated as the program progresses and experience accrues, will be key to ensuring that decision making at all levels is not delayed because one of the following is lacking: (1) the information needed for VV&A of AI/ML datasets and models or (2) the T&E results that demonstrate that the system's design and performance adhere to the DoD AI Ethical Principles and the DoD Responsible AI Strategy and Implementation Pathway (see Sections 2.2.1, 2.2.2, and 2.3). Producing an initial version of the IDSK to support the pre-MDD review of an AIES/MLES program will likely be necessary to gain approval, as the IDSK will be essential to identify T&E

[7] These DoD issuances and guidebooks are under development. This guidebook will be updated with specific citations when these documents are complete and approved.

resources, including models and infrastructure, that do not exist and need to be developed to ensure adherence to DoDD 3000.09 requirements.

2.2.4.3 Operational Test Readiness Review

Authorities and Responsibilities

In accordance with forthcoming DoDI 5000.DT, DT&E activities include validating system functionality in a mission context to assess readiness for OT. The OTRR process will include a review of DT&E results; an assessment of the system's progress against the KPPs, KSAs, and critical technical parameters in the TEMP or other test strategy documentation; an analysis of identified technical risks to verify that those risks have been retired or reduced to the extent possible during DT&E or OT&E; a review of system certifications; and a review of the initial operational test and evaluation entrance criteria specified in the TEMP or other test strategy documentation. the TEMP constitutes general guidance for whatever organization is responsible for overseeing DT&E and advising decision authorities on the readiness to conduct OT. At the level of the Office of the Secretary of Defense (OSD), the USD(R&E) (through DTE&A) provides program assessments for ACAT IB/IC programs on the T&E oversight list to support the OTRR. The program assessment is based on the completed DT&E and any OT&E activities completed to date and will address the adequacy of the program planning; the implications of testing results to date; and the risks to successfully meeting the goals of the remaining T&E events in the program.

AI-/ML-Specific Operational Test Readiness Review Considerations

As is the case for any defense system, much of the information needed to conduct the OTRR will be system specific. However, new issues and concerns exist with policy compliance and performance certification specific to AIES or MLES. Plans for obtaining the information necessary to address these issues should be made at the outset of the program.

Risks

Evidence will be reviewed to determine the extent of any substantial remaining risks for compliance with the requirements specified in DoDD 3000.09, including compliance with the DoD AI Ethical Principles. Regarding the latter, the areas likely to be of particular interest include whether the AIES/MLES has operational methods that are well understood; whether its uses/ or CONEMPs are well-defined and understood; and whether unintended behavior can be readily detected and stopped (see the "Safety Documentation" section below).

Review of the DT evidence relevant to all the other AIES/MLES-specific risk characterizations discussed in Section 2.2.2 should be anticipated.

Performance

Review of the DT evidence relevant to all the performance characterizations discussed in Section 2.2.1 should be anticipated. Whether emergent behavior has occurred, and if so, in what regions of the potential operational space it occurred and its causes, will be of substantial interest. If that behavior was potentially detrimentalor unexplained, then the provisions made for conducting OT so that the particular behavior should not occur, and can be promptly detected and halted if it does (see the "Risks" section above), will need to be described in detail. Any significant changes in performance observed associated with variations in test conditions will be of interest, particularly if the changes were detrimentalor the variations in test conditions can be characterized as small.

Safety Documentation

Documentation (e.g., Safety Release, Safety Confirmation, Safety Certification) will be required to conduct OT using the operational Service personnel that will employ the system when it is fielded. Assurance case documentation (e.g., Tate 2021) and information collected during DT will be the basis for obtaining those certifications. The safety evaluator is responsible for generating the Safety Release in support of testing involving Service members as well as the Safety Confirmation in support of equipping, fielding, and acquisition milestone decisions and for providing safety input to the evaluation reports. Typically, the test manager is the safety evaluator. Testers should collaborate with the Services' safety authorities throughout development to ensure they have, and were promptly provided with, the information required to obtain needed safety documentation.

It is likely that safety authorities will be particularly interested in evidence of compliance with the attributes of the DoD AI Ethical Principles discussed above as well as in whether emergent behavior has occurred. The need for evidence of the ability of operational personnel to identify and promptly stop unintended or detrimental behavior and evidence of complete understanding by operational personnel of the AIES/MLES CONEMP should be anticipated.

Other Certifications

Review of the status of and evidence supporting cybersecurity, joint interoperability, and any mission-specific certifications (such as airworthiness) should be anticipated (see Section 4.6). Regarding interoperability in particular, to the extent AIES/MLES will conduct and control interactions with other linked systems during OT, the need for evidence supporting the proper functioning of that interoperation without damage to or corruption of the interacting, linked systems should be anticipated.

Pre-Fielding Review

As discussed in Section 2.2.2.1, a pre-fielding review of compliance with DoDD 3000.09 will be conducted for any AIES/MLES. OT will provide data on the system's operation under the most operationally realistic conditions in which the system has been employed. Therefore, it should be anticipated that the plans for collecting data during OT that will support a successful pre-fielding review will be a topic of interest.

2.2.4.4 Contract Structures and Data Rights

It is important that the T&E community be prepared to articulate the value of data and software access. The contract and contract structures used in an MLES acquisition will control the government's data rights. Government data rights will have enormous effects on what is possible and how to proceed with the T&E of ML components or the MLES. "Data rights" refer to both technical data and computer software. It needs to be understood that for ML systems, "technical data" refers to TVT data used by the contractor in developing the ML model as well as model performance data. Given the importance of the full development history in assessing MLES, elements of the diagnostic data used by the contractor in development may be of value as well.

The Test and Evaluation Strategy (TES) will be greatly affected by which data the testers can access. Early attention is essential to optimizing access to the data relevant for T&E. Broad Agency Announcements can apply for funding under budget activities 6.1 (Basic Research) through 6.4 (Advanced Component Development and Prototypes) and allow for flexibility rather than precise performance standards. This flexibility is of particular value in early exploration of competing ML approaches. The close coupling and iterative testing of the design and the CONEMP together is well suited to exploration within flexible contracts.

RFPs are a means to communicate government requirements to contractors early in the acquisition life cycle. The specificity of requirements and, under some circumstances, the specification of evaluation criteria are potentially problematic for ML development. Premature specification of evaluation criteria can interfere with the needed iterative natures of both the development and the testing of MLES.

The default language currently in use concerning data and software was generally written without consideration of the recent surge in ML applications. Default language on "data" may not include TVT data. Similarly, default language on "software" may not include the ML model. In addition to contract structures, attention needs to be paid to the specific language on data and software. Again, it is essential to articulate the benefits of data and software access, so the benefits can be weighed against the costs.

A major distinction in contract structures is between Federal Acquisition Regulation (FAR) and non-FAR contracts. The FAR and the Defense Federal Acquisition Regulation Supplement (DFARS) carry a presumption that the government will generally not be willing to pay for data. FAR-based contracts may warrant early and careful attention to the provisions for data rights. (For FAR-based contracts, there are eight types of data rights, described in AcqNotes, The Defense Acquisition Encyclopedia.)

Non-FAR contracts are "negotiation based" and will generally offer more flexibility in obtaining access to data or software. For more information see the AI Acquisition Guidebook.

2.2.4.5 Schedule Estimation

Part of the purpose of test planning is to support accurate estimation of the cost and duration of system development. For traditional system development, this estimation is typically accomplished through the development of an Integrated Master Schedule that accounts for all of the work elements to be completed, their sequential dependencies, and their anticipated durations; for more information, see the Integrated Master Plan and Integrated Master Schedule Preparation and Use Guide. The resulting project network can be analyzed in terms of critical paths and minimum or maximum anticipated time to project completion. This approach, however, assumes that the set of work elements needed to complete the project is known in advance and that each work element will be executed exactly once.

For MLES, this assumption will in general not be justified. Developing and training ML models to support defense missions is typically an iterative process, in which multiple successive models are trained before a satisfactory model is found. This iterative process might involve changes in the kind of AI model being trained, the TVT data used in training, or even the software architecture in which the model is embedded. As a result, inherently more uncertainty exists in the duration of the development effort than there would be for typical software development. For programs using agile software development approaches, the impact of ML on the schedule depends critically on whether the ML-enabled capabilities are a necessary part of the minimum viable capability release or are optional capabilities that might be added in later sprints.

Use of ML can also affect the development schedule if testing will require access to specific test resources, such as instrumented testbeds, systems integration (SI) labs, or specific personnel. Obtaining timely access to test resources becomes even more challenging when development schedules are uncertain because of iterative design (and redesign) cycles for the ML modules.

At a practical level, test activities identified in the IDSK should be characterized by whether they are expected to be one-time events, iterative events, or ongoing near-continuous assessments.

This characterization will help to inform PMs and decision makers regarding likely schedule impacts due to ML-related testing.

2.3 CDAO T&E Strategy Frameworks

Many organizations are producing frameworks and guidance for the emerging field of T&E of AI, such as the MITRE Technical Report, "Systems Engineering Processes to Test AI Right (SEPTAR) Release 1" (Balhana et al. 2023) and the CDAO T&E Strategy Frameworks (see Guidance on the CDAO Joint AI Test Infrastructure Capability (JATIC) Documentation Website. The CDAO frameworks, created by the Assessment and Assurance Division, discuss AI T&E guidance in four focus areas: model T&E; human-systems integration (HSI) T&E; SI T&E; and OT&E, which is referred to in Section 2.3.4 as mission-informed T&E. In Source: CDAO T&E Strategy Frameworks

Figure 2-3, these focus areas are represented as puzzle pieces in the framework: OT&E is represented as blue, HSI T&E as orange, model T&E as green, and SI T&E as yellow. These four areas are not mutually exclusive and realistically would overlap in T&E. The material presented in the framework is not exhaustive and is designed to help testers understand T&E concepts for writing and assessing the TES. The AI T&E framework not only introduces new concepts and methods but also encourages a change in traditional T&E methods by pushing some test aspects earlier ("shifting left") and extending other test aspects past deployment into post-fielding ("shifting right"). The four focus areas of the CDAO T&E Strategy Frameworks are discussed in the following subsections.

Source: CDAO T&E Strategy Frameworks

Figure 2-3. Four Focus Areas of the CDAO T&E Strategy Frameworks

2.3.1 Model Test and Evaluation

AI models must undergo T&E to determine model performance, especially in mission-realistic environments. An AI model TES depends on the model’s learning paradigm, which can be SL, UL, RL, or some combination of the three. Regardless of the learning paradigm, the overall T&E process is unchanged. This process involves designing an appropriate test plan; selecting and curating datasets for training the model; selecting, training, and testing an appropriate model; and keeping rigorous documentation throughout the process.

CDAO recommends that documentation occur over the entire AI model T&E, as it helps ensure that the AI model is comprehensively tested for key use cases and operational contexts; is monitored for drift; and is as transparent and understandable as possible. Documentation should include, at a minimum, model characteristics, data sources, data preprocessing methods, modeling techniques, and evaluation metrics. Implementation of data cards and model cards allows for comprehensive documentation. More information on documentation as well as data and model cards can be found in Sections 2.2.2 and 2.2.3.3.

Testing an AI model determines its quality, reliability, and usefulness. Correctness is an intuitive metric, but other aspects such as bias, explainability, robustness, and resilience can ensure the quality and reliability of the model (e.g., see Pullum 2022 and Section 3.2).

Implementing the TES requires a test plan with the appropriate test types and designs. Some test methods for testing model parameters include testing by comparison to another model, such as A/B testing or back-to-back testing; testing for adversarial threats, such as red teaming or other adversarial testing; experience-based testing that relies on the tester’s experience; and metamorphic testing, which focuses on the relationship between inputs and outputs. Test design also requires choosing appropriate test factors, which can be difficult for AIECs as some factors are not easily identifiable because of the “black box” nature of some AI algorithms. Subject matter experts (SMEs) may help in identifying these factors. Common test design methods include cross-validation, holdout validation, and bootstrapping (e.g., Cai et al. 2023).

The CDAO T&E of AIEC Framework stresses that data are the foundation of an AI model’s performance. Data can come from many places, including commercial datasets, open source data, lab environment data, and real environment data. Certain steps can be taken to ensure data quality across the life cycle, including performing V&V throughout the entire process. The first step is data selection and acquisition; data should be abundant and as operationally realistic as possible. Data should then be cleaned of entry errors, corrupted information, duplicates, and other quality issues discovered. The data should also be clear of any extraneous information that may skew the model’s performance. The data should then be formatted for input into the AIEC and split into subsamples of training data, validation data, and testing datasets. The data should be curated, organized, and documented.

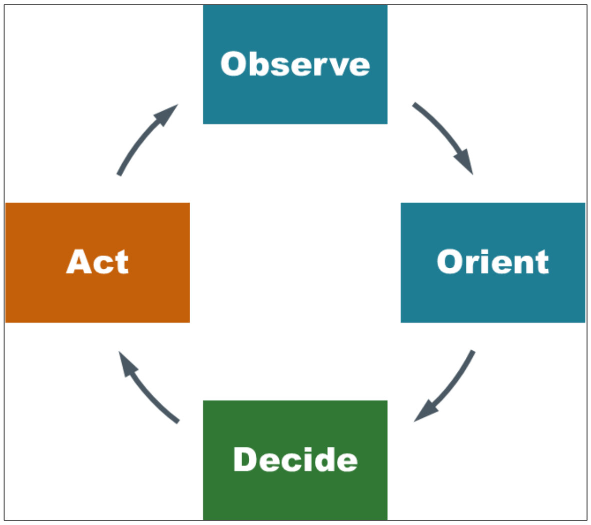
The AI model life cycle has six steps. The first step is selecting the optimal architecture. The model should be of the appropriate complexity and fit into any operational hardware constraints. The next step is defining the data features that will be used by the model, performance metrics, and model interpretation methods. The model can then be trained and validated using the training or validation data. Next, the model is evaluated using the set-aside test data to determine whether it meets mission objectives and requirements. Finally, the model can be deployed and should be continuously maintained.

Environment constraints are another consideration for AI model T&E. The available computing power, network connectivity, and security limitations should be taken into account when planning the test approach. The classification level, in particular, can place requirements on how data are handled and shared, which may affect the data available for training, V&V, and testing.

2.3.2 Human-Systems Integration Test and Evaluation

HSI T&E ensures that warfighters can efficiently, effectively, and safely leverage AIECs to accomplish tasks. HSI discussions must cover surface-level system features, such as ergonomics, as well as more operational features, such as usability and workload. HSI evaluations should consider how factors that affect the system also affect the human user, and how factors that affect the human also impact system performance. By understanding how users mediate the effect of operational factors on system performance, testers will be able to prioritize HSI concerns.

Some personnel in DoD are familiar with Boyd’s observe-orient-decide-act (OODA) loop, shown in Figure 2-4, which is used to determine how to complete a mission’s objective. CDAO asserts that the OODA loop is a useful basis for HSI T&E because most HSI concepts directly relate to the performance of one or more of the decision-making stages of the OODA loop. Novel AIECs, however, will make HSI T&E more complex. AIECs will perform all of the steps in the OODA loop, and teaming systems will have interactions across the loop. The CDAO HSI T&E framework discusses 13 distinct HSI concepts that impact the warfighter’s ability to meaningfully leverage their AIEC. The following discussion outlines these 13 concepts and expands on a few key concepts. Table 2-3 lists the 13 concepts and recommends TES actions for each concept.



Source: CDAO HSI T&E of AIECs

Figure 2-4. Boyd’s Observe-Orient-Decide-Act Loop

The first seven HSI concepts relate to the warfighter’s need to understand their environment and system. This need falls into the OODA “Observe and Orient” phase. The HSI concepts are mental models; boundary awareness; SA; information quality: objectivity, utility, and interpretability; and XAI. XAI has many definitions; in the CDAO framework, it refers to how an AIEC provides the “why”—the causal reasons for an AIEC’s internal logic and resulting output—in a way that human operators can understand. Good XAI explanations are useful for deciding whether the AIEC performance will be consistent. For a human operator or warfighter to better calibrate their trust in and collaboratively problem solve with an AIEC, XAI methods should be developed for operators to understand the system’s decision-making process (see Section 3.5.3).

The next three HSI concepts relate to the OODA “Decide” phase, when the warfighter needs to decide whether or not to use the AIEC. These concepts are trust and reliance; emergence; and workload. Trust is the belief that an AIEC can be depended on in any situation, especially in vulnerable or uncertain situations. The critical outcome of trust is reliance, which is the use of the system in those situations. The warfighter’s trust is properly calibrated when their reliance on the system matches the system’s performance. The TES should address warfighter trust across operational conditions and evaluate calibration relative to system performance (see Section 2.2.3.42.2.3.4).

The final three HSI concepts relate to the OODA “Act” phase, when the warfighter needs to govern and monitor the AIEC to carry out the warfighter’s intent. These concepts are function allocation; usability; and training quality. Usability is the fitness of a tool for a task and is composed of utility and ease of use. Even in an AIEC that autonomously executes tasks, humans will still need to interact with the AIEC to give initial orders, extract information, or potentially intervene to alter or stop its behavior. AIECs that execute actions need to be assessed on whether those actions match the operator's intent.

Table 2-3: Summary of Recommended TES Actions for Various HSI Concepts

OODA Phase	HSI Concept	TES Actions
Observe and Orient	Mental Models	Assess the content of the warfighters' mental models and evaluate how well the warfighters' models allow the warfighter to predict AIEC system behavior.
	Boundary Awareness	Evaluate the warfighters' knowledge of AIEC system limitations.
	Situational Awareness (SA)	Employ environmental SA measures beyond self-report. The TES should not commit to this if adequate resources will not be assigned.
	Information Quality: Objectivity	Compare the accuracy and uncertainty of the information provided versus warfighters' needs across operational conditions.
	Information Quality: Utility	In both DT and OT, test the usefulness of the information provided for successfully completing warfighters' tasks.
	Information Quality: Interpretability	In OT, measure whether information is communicated in an understandable way and under operationally realistic workload spikes.
	Explainable AI (XAI)	Identify which XAI definition was adopted for the test and measure the effect of AIEC system explanations on mission performance and warfighter decision making.
Decide	Trust and Reliance	Measure warfighter trust across operational conditions and evaluate calibration relative to system performance.
	Emergence	Resource free-play testing where emergence can arise from all agents, and follow up on any emergent behavior.
	Workload	Measure nominal workload as well as off-nominal workload within safety constraints.
Act	Function Allocation	Require programs to submit a function allocation for evaluation as part of the assurance case for the system.
	Usability	Evaluate usability at a granular subsystem level for DT, and holistically examine the system of systems in OT.
	Training Quality	Assess training quality on representative warfighters—not engineers, contractors, or "golden" crews.
Note: To accomplish each TES action, CDAO lists "state-of-the-art measurements" comprising behavioral tests and benchmarks, surveys, and qualitative user interviews. In several cases, however, CDAO notes the need for development of new survey scales. In two instances (XAI and emergence), CDAO notes that there are no "off-the-shelf" measurements available.		

Source: CDAO HSI T&E of AIECs

Over the AIEC life cycle, CDAO asserts that some HSI aspects can be shifted left and others shifted right. HSI design choices should be tested early and often. Testers should put AIECs in front of operational users early and often to ensure that the technology is usable and understandable. Early interactions with users can help develop operational concepts and mock-up user interfaces and can help ensure that the AIEC is developing capabilities that help users accomplish their missions. On the other hand, some T&E aspects should continue after the AIEC is fielded. As fielded AIECs interact with each other and with the warfighters during operations, emergent behavior may increase. The TES should address monitoring and mitigating emergent behaviors and any associated resources required. Additionally, warfighters often lose engagement with their work as AIECs take over tasks, so the TES should consider post-fielding evaluations of both the AIEC and the warfighters. Warfighter training programs should also continue post-fielding and adapt to account for any changes in the AIEC’s behavior over time.

2.3.3 Systems Integration Test and Evaluation

SI is a structured approach to combining hardware and software components into a functional system. An interconnected, integrated system is more efficient than multiple stand-alone systems and can reduce cost and redundancy. SI can be daunting and error-prone if executed all at once, so CDAO recommends iteratively integrating AI components or AIECs into the larger system and frequently testing for errors or issues. Additionally, the scope can be kept manageable by implementing robust and modular architectures and efficiently designing testing.

Before beginning integration of an AI component into a system, the tester should ensure that the system is stable and that the AI component has completed stand-alone testing. The stand-alone testing should provide a comprehensive characterization of the AI component and should include T&E using operationally realistic synthetic or collected input data. Full characterization may be useful in solving any integration issues by being able to adjust the component to suit the system. Understanding the AI component allows for educated adjustments rather than blind changes.

SI test objectives include characterizing the performance of the hardware, software, and AI component; identifying and documenting changes in performance between iterations; providing data showing that the system fulfills the requirements; and using the test results to create and maintain a system assurance case (e.g., see Tate 2021; Hawkins et al. 2021). The CDAO T&E of AIEC Framework introduces five core T&E activities for SI: functionality, reliability, security, compatibility, and interoperability. This list is not exhaustive, though testing these characteristics can demonstrate whether the AIEC fulfills the system’s requirements. SI is also an iterative process—analysis may show that the AI component needs to be retrained, and if so, then the component should be tested again after retraining.

Functionality is the ability of the system to do its intended work and meet the specified requirements and standards. Though a component may be functional on its own, unanticipated behaviors and relationships may occur when system components are integrated. This is called intra-system emergence. Catching and mitigating emergence are most easily done by incrementally integrating and testing components. Testers should identify relevant metrics and criteria for each integration step, and test design should consider parameters outside of the AI component's training data, such as changes to other system hardware and software. Test methods should also include edge cases that may increase the chance of emergent behaviors. Data collection should be sufficient to analyze the AI component and system performance, which will demonstrate whether the component needs to be retrained.

The CDAO SI T&E of AIECs framework defines reliability as the probability that a system successfully performs a function under stated conditions for a stated period of time. It can be difficult to predict or evaluate which operational conditions are likely to cause failures, especially because AI components do not physically degrade over time. However, AI components may experience performance drift, where the performance degrades over time either through explicit learning or changes in operational inputs.

Security testing identifies and attempts to mitigate a system's vulnerabilities and weaknesses and demonstrates whether security requirements have been met. Testers can reference the DoD Cybersecurity T&E Guidebook to learn more about cybersecurity requirements and T&E guidance. Varying security levels across different components can create weak links, providing easier opportunities for attackers to exploit these inconsistencies. A vulnerability in one component can lead to cascading failures. Component interactions may also produce unpredictable, emergent behaviors that result in new vulnerabilities that could be exploited by adversaries or adversarial AI. "Adversarial AI" is AI that targets other AI systems through poisoning, evasion, or privacy attacks. To effectively test the security of an integrated system, CDAO recommends that testers utilize a system threat model to identify potential threats and corresponding failure modes; develop test cases to ensure system security; and include automated testing to find new vulnerabilities and analyze performance. Such models could be of considerable value, but the expertise and cost of developing and upgrading them are usually considerable.

Compatibility is how well two or more components interact together. Compatibility testing also considers whether the system can function in the operating environment and assess whether an AI component can consistently function across various operating systems and fielded environments. New software updates, hardware changes, or other system component changes could all affect the compatibility of an AI component and may require that the component be retrained. Compatibility between multiple AIECs may require training them together.

Interoperability testing of an AI component should verify independent functionality and seamless communications between components and should account for the AI component's dynamic nature. Because the performance of AI components relies on the quality and relevance of training data, AI outputs change as new versions are trained on updated datasets. Ensuring compatibility between different component versions is crucial. Testers should analyze the change in AI component behavior during integration and consider how the AI component handles input from other system components.

Over the AI T&E life cycle, effective SI will require both shifting left to anticipate mission-level testing needs early in development and shifting right to continue DT activities during operations and sustainment. Testing must be done early and often to identify and correct integration issues. Documentation should also begin early to identify edge cases; track training datasets and environments; and maintain version control on both the AI component and the training datasets. Post-fielding, performance in the field should be monitored; additional testing may be needed to evaluate the retrained AI model. The AI component may be brittle and lose effectiveness when faced with new environmental conditions, such as a dirty camera lens or a new lightweight hardware component. Any deviations from the expected behavior should be identified, and a process should be put in place to determine when these deviations require intervention of some kind.

2.3.4 Operational Test and Evaluation

CDAO discusses OT&E, which, in its definition, encompasses any testing conducted under realistic mission conditions across the acquisition life cycle, including formal OT&E. Realistic mission conditions are essential for the development of a robust system, with understood performance that can be generalized to other operational conditions. This section focuses on the CDAO discussion of the need for realistic T&E of AIECs beginning as early as possible and continuing through the life cycle, including during development, which is referred to as mission-informed T&E. Behavior of an isolated AI model or an AIEC in a lab test is likely to be significantly different than in real, operational use. Mission-informed testing allows for sufficient opportunities to catch and mitigate deficiencies that can reveal system limits. Mission-informed T&E must cover an expansive scope and requires a nuanced test design that uses DOE methods (Montgomery 2020).

Most challenges associated with mission-informed T&E of AIECs are not unique to AIECs, but the nature of AI can complicate problems that testers already face. AIECs can perform unpredictably when deployed outside of the conditions in which they were trained simply because their ML models cannot deal effectively with relationships not explicitly seen during training and because of the potential for emergent model behaviors with sensitivity to small changes in input. The CDAO OT&E of AIECs framework discusses five main challenges for mission-informed testing that are exacerbated by AIECs: resourcing test activities; generalizing from test to field; characterizing causal relationships; detecting and mitigating novel threats; and tracking performance drift.

Mission-informed T&E is constrained by limited resources, such as time, money, staffing, and infrastructure. Efficient DOE is important yet difficult for AIEC systems because these systems require more test resources than traditional systems. The frequent changes in AIEC behavior demand more frequent testing to measure performance accurately. Additionally, the OT envelopes for AIEC systems are larger and have unpredictable performance that may require unconventional DOE approaches, such as prioritizing edge cases and low-density training data regions. Many factors for DOE in AIECs are unknown, requiring more data to determine these factors. The larger test envelope significantly impacts test resources, as many programs lack sufficient time, money, or personnel. To address these challenges, CDAO suggests that testers may need to develop new methods and metrics to measure and evaluate AIECs. Implementing productivity-boosting measures, such as automated testing, could help mitigate the constraints of limited resources.

Testers use generalized T&E results to predict fielded performance, but an AIEC’s ability to identify subtle trends can make it challenging to identify appropriate test factors. Emergent behaviors also make generalization difficult as this change in AIEC behavior is difficult to predict. Additionally, the understanding of the data features that influence an AIEC’s performance is less mature, especially for non-industry applications. If an AIEC is trained with poor-quality or nonrepresentative data, it will likely be ineffective, but it may appear effective if the test dataset is also not operationally realistic. To make accurate generalizations, operational realism should be introduced early in AIEC development. Testers may need to find ways to make test ranges operationally realistic for AIECs.

Understanding the causal relationships between inputs to the system and performance outputs helps in predicting system performance in untested scenarios. Characterizing causal relationships for AIECs, however, is especially difficult. This challenge is exacerbated by the black box nature of algorithms in AIECs, which obscure internal decision-making processes. The CDAO OT&E of AIECs framework states that the AI test community needs new T&E methods for AIECs: Current T&E methodologies are insufficient because they assume system performance can be predicted based on outcomes from a small number of test points, which is not true for AIECs. Using assurance cases to scope a set of possible failure modes is one approach to this challenge (e.g., Tate 2021; Hawkins et al. 2021).

New technology such as AIECs introduces new attack vectors that must be detected and mitigated. Systems using AIECs may have large cyberattack surfaces, and AIEC-specific vulnerabilities should be addressed and mitigated. These vulnerabilities include extraction, where adversaries query an AIEC to reverse-engineer sensitive information, and data poisoning, where adversaries tamper with the training datasets. Additionally, the autonomous nature of some AIECs increases the risk of unplanned behavior slipping past human operators. Testers should create adversarial examples and use other robustness training techniques during AIEC test and development. They should also develop metrics to assess whether AIEC features are likely to decrease or increase vulnerabilities. Additional information on attack surfaces and cyber DT&E of AIECs will be addressed in the forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook.

Over time, AIEC fielded performance can diverge from tested performance—a phenomenon known as performance drift. Several factors contribute to AIEC performance drift. Changes in the operational environment can lead to input data drift, affecting performance. Additionally, a brittle or overfit AIEC may perform poorly in operationally realistic conditions where inputs differ from the training data. Drift may occur as the AI model learns and exploits the system's internal reward function. Drift can also be caused by frequent AIEC software updates. CDAO recommends that testers monitor performance drift by continuously measuring system performance, especially after deployment.

Throughout the AI T&E life cycle, certain T&E aspects should be adjusted. Early DT should prioritize operational realism and mission relevance to better understand AIEC performance and behavior in a mission context. Testers should have early access to and knowledge of the training data to ensure that the dataset is operationally representative and to document any changes in the training dataset. M&S can help early AIEC testing to be operationally realistic, though currently M&S applications are not yet ready for mission-informed testing of AIECs. M&S that has been through VV&A for conventional systems will likely need to go through a new VV&A process for use with AIECs. Once an AIEC is ready for deployment, fielding it in phases such as a limited capability rollout or selective user testing can reduce risk. Post-fielding, testers and users should establish performance thresholds that trigger intervention for AIECs. Consistent and continuous system performance monitoring will indicate whether changes in the operational environment or to the AIEC itself require retraining or additional T&E.

2.4 Implications of AI for DT&E Across the Life Cycle

Capability assessments alone are insufficient for T&E of AI-based systems. The complexity and uncertainty of AI models require T&E to both encompass and inform the processes and methodologies used to create these AI models.

This section addresses the consequences introduced by AI across the T&E life cycle, with a focus on ML models.

2.4.1 The ML Development Pipeline and Its Consequences for T&E

The ML pipeline depicted in Figure 2-5 highlights several significant departures from the development pipeline typical of conventional non-ML software tools. Unlike most non-ML development pipelines, the collection and preparation of suitable high-quality data are central to the ML pipeline, supporting both the training of the ML model and its assessment. Figure 2-5 illustrates the typical ML pipeline, which begins with the requisite data collection and preparation for model training. Once sufficiently trained, the model is prepared for operational use and then deployed. Each of these steps warrants careful validation to ensure desired behavior, especially in applications for which safety is vital.

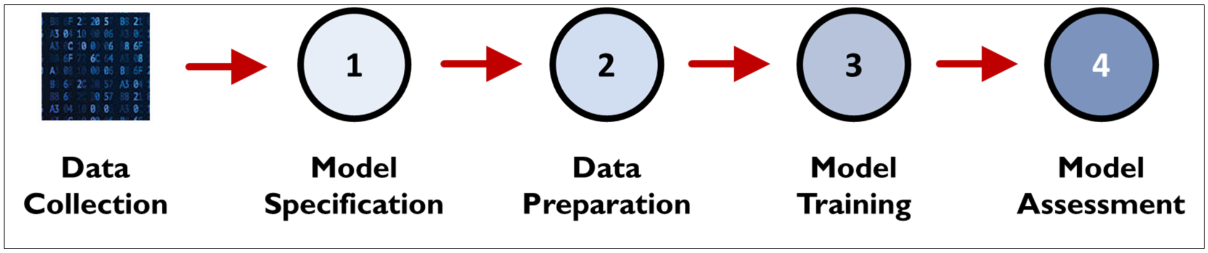


Figure 2-5. Typical Machine Learning Pipeline

Despite the large variety of ML approaches and models, the overarching development pipeline consists of the same four essential phases, which are illustrated in Figure 2-5 and described below:

1. **Model Specification**: Developers must make key decisions, such as determining the necessary input data, desired outputs, and metrics for assessing success.
2. **Data Preparation**: Developers need to format input data, identify and remove flawed data, and manage outliers. For supervised models, developers must also assess the suitability of data labels.
3. **Model Training**: Data are divided into training and validation (T&V) datasets. The ML model learns from the training dataset, while the validation dataset is employed to assess and guide the model’s learning algorithm.
4. **Model Assessment:** The trained model is evaluated using a test dataset to assess its performance.

Given the importance of data in the ML pipeline, datasets for TVT compel new considerations for V&V, while the test strategy and planning must account for transparency, robustness, brittleness,[8] and adversarial vulnerability. Additionally, ML models introduce risks in HMT, such as unforeseen emergent behaviors and failure to account for automation bias or other factors

[8] The brittleness of an ML model concerns its sensitivity to small changes in its inputs.

affecting operator trust in model outputs. See Section 3.2 for a more detailed discussion of the pipeline.

2.4.2 T&E Considerations of ML-Related Risks

To ensure the appropriate consideration of ML-related risks and the pursuit of necessary remediated steps, a cogent risk-management strategy must guide the T&E process; for more information see the NIST AI RMF 1.0.

According to NIST, a precondition of assessing risk entails an understanding of the mission context, in which T&E teams identify threats to safety and establish apt metrics for gauging performance. Potential risks must be carefully scrutinized and tracked to accurately characterize the capabilities, constraints, and vulnerabilities of the ML models. T&E teams should then inform the eventual users of ML-based systems about the trade-offs between risk and performance.

2.4.3 ML Resource Planning for T&E

The inner workings of many ML models are poorly understood, even by their developers. This lack of transparency not only hinders development and T&E efforts but can reduce trust in ML outputs. Thus, T&E for ML-based systems will require new resources, such as simulation software for testing; appropriate training for T&E teams and operators; and tools that can characterize the internal decision-making processes of ML models. Although T&E will often call for expanded access to conventional equipment, resource planning must account for potentially protracted training timelines and address the safety requirements associated with the specific ML model and mission context.

2.4.4 Test Strategy and Planning

The general content expected in a TES, TEMP, or other analogous planning document for any program is described in Paragraph 4.1 of forthcoming DoDI 5000.DT. Test planning should begin at the earliest point, starting in mission engineering, science and technology, prototyping and experimentation, and any development for eventual insertion into DoD networks, systems, platforms of systems, and system-of-systems. Timely updates to the planning documents should be made to support decision making and preparations for test execution and evaluation as the program progresses. In accordance with forthcoming DoDI 5000.DT, key DT-related information expected in test planning documents includes the following:

- An integrated test program summary and master schedule.
- DT events for evaluating performance, interoperability, reliability, and cybersecurity.

- Needed test infrastructure and tools (e.g., models, simulations, automated tools, synthetic environments) and the strategy for conducting the VV&A of those tools.
- Complete resource estimates for all T&E to be conducted and associated tools and infrastructure.
- An IDSK tabulating expected decisions and the information needed to support them.

2.4.4.1 Key Aspects of Test Planning Specific to AIES/MLES

This subsection assumes that most, if not all, programs incorporating AI/ML will go through the equivalent of the TMRR phase (hereinafter referred to as early development) and EMD phase (hereinafter referred to as detailed design and development) of an MCA (see Section 2.2.4.2). Test planning post-fielding will also be discussed. The issues for DT associated with the pre-program initiation and pre-fielding reviews mandated by DoDD 3000.09 are not discussed here; see Sections 2.2.4.2 and 2.2.4.5 for discussion of those reviews.

Early Development

In early development, efforts will likely focus on data collection, model specification, and data preparation. Plans for evaluating the adequacy of the data needed to train, validate, and test whatever ML method is selected will be made and executed, at least in part. (See Sections 3.2 and 3.3 of this guidebook for discussions of some of the associated considerations.) A key consideration will be measurement of the similarity and dissimilarity among the TVT datasets as they are iteratively developed. Although similarity of test and training datasets is usually regarded as desirable, excessive similarity could render any evaluation of the ML model inapplicable to conditions it was not explicitly trained to but will nonetheless encounter. Conversely, excessive dissimilarity could cause the ML model to fail without substantively informing the evaluation. Metrics for assessing similarity and dissimilarity are current research topics and include surprise adequacy, combinatorial coverage, and set difference combinatorial coverage (Chandrasekaran et al. 2023; Lanus et al. 2021).

Evaluation of the consistency of the TVT datasets with the conditions and characteristics of the ML system’s expected operating environment and CONEMP will also be needed. However, metrics for this evaluation are generally lacking, and SME judgment will likely have to suffice until the results of T&E conducted during detailed design and development are available. This implies that the steps in the ML pipeline depicted in Figure 2-5 may need to be accomplished repeatedly and iteratively during early development as well as detailed design and development. Test planning should attempt to account for, and will have to be revised as needed to reflect, expected iteration. Evaluation of data-related issues including compliance with DoDD 3000.09

and ensuring a lack of bias in the data will also be necessary. Additionally, assessing the potential for data poisoning to have occurred will also require T&E.

Testers can aid model selection and specification in at least three ways:

- By informing the PMs and design teams of past experience and the results of applying particular ML methods and approaches to analogous operational conditions and CONEMPs.
- By informing design teams and PMs of the test infrastructure and tools (and the associated resources and schedules) needed to implement the ML methods under consideration. Estimates of resource needs and schedules should include those needed to conduct VV&A of the infrastructure and tools as well as to conduct VV&A of the ML model itself as well as the integrated system (see Section 3.2.5 and Sections 3.4 through 3.6).
- By planning for, conducting, and evaluating early iterative testing of representative data used to train alternative prototype ML models.

As the design team makes decisions on the ML method(s) to use, detailed planning should begin for acquiring any new tools and infrastructure; conducting VV&A of all the acquired tools and infrastructure; characterizing the performance and risks of the ML model as a component of the system; and subsequently characterizing the performance and risks of the ML model as integrated into the overall system (see Sections 2.2.1 and 2.2.2). Although the last three activities may be conducted primarily during detailed design and development, planning for these activities and obtaining the needed resources should begin during early development.

Instrumentation needed for assessing an AIES/MLES will likely need to be an integrated part of the system. Testers should work with designers beginning in early development to specify the information needed for T&E of the system and to enable operator trust in the system by providing real-time explanation of the ML's operation and decision making (see the Defense Science Board Summer Study on Autonomy for more information). Special tools and infrastructure that may be needed to collect, store, and analyze data generated by that instrumentation should also be identified and resourced. Additionally, any special tools and infrastructure (such as person-in-the-loop simulations) that may be needed to evaluate HMT (see below) should also be identified in test plans as early as possible.

Detailed Design and Development

Detailed design and development efforts will likely focus on refining the selection of the ML model(s) to be used; model training and model assessment (including VV&A of the ML model itself); as well as VV&A of tools and infrastructure. Initially, the performance and risks of the model(s) will be assessed as a component of the system. Subsequently, the performance and risks of the overall system with the integrated model(s) will be assessed. Training and assessment will be conducted iteratively and likely repeatedly as issues in performance are discovered through T&E of both the stand-alone ML model(s) and the integrated system. Therefore, test planning and documentation will need to be iterative and dynamic. Government testers' involvement in conducting independent T&E will need to be defined contractually.

Metrics for assessing the ML model itself as a system component, as well as for assessing the ML-specific issues associated with the integrated system, are areas of active research. Commonly used metrics, when ground truth is available to evaluate performance of an ML model on a test set, include accuracy, precision, recall, and F_1 score for classifiers. Root mean squared error, MSE, and mean absolute error (MAE) are used for regression-based ML models (Chandrasekaran et al. 2023; Hutchinson et al. 2022). The most appropriate metrics will depend on the mission, environment, and identified COIs.

Other potential approaches that could be incorporated in test planning for evaluation of the ML include metamorphic testing,[9] differential testing, combinatorial testing, fuzz testing, and adversarial testing. Metamorphic and differential testing address the lack of ground truth by evaluating whether the character of model responses to inputs is reasonable, while combinatorial testing attempts to comprehensively cover a large operational space (Chandrasekaran et al. 2023). Adversarial testing is the ML-specific analog to testing the performance of an integrated system when operating against threat attacks.

T&E of the overall performance requirements of the integrated system must also be planned and conducted (e.g., the ability of an AI-/ML-enabled combat aircraft to successfully penetrate threat air defenses and destroy targets). This testing can use some of the more traditional metrics and methods for evaluating the performance of DoD non-AIES/MLES but must account for the issues described in Sections 2.2.1 and 2.2.2 specific to AI/ML, such as compliance with DoDD 3000.09, including assurance that human intervention or runtime monitoring and intervention can stop the system from behaving in an unexpected or harmful manner; robustness (e.g., Hendrycks and Dietterich 2019); brittleness; and emergent behavior. The need for an information-enabling evaluation of these aspects of AIES/MLES behavior, which can be highly nonlinear as a function of the characteristics of the operating environment, means that the coverage of test points in the system's operating environment may have to be denser and wider than the test plans generated using the optimality approaches typically employed in methods such as DOE.

[9] Metamorphic relations involve testing for ML model behavior that should be invariant, decreasing, or increasing depending upon changes in input. For example, the output of a natural language processor should not change if a synonym is used. See https://www.giskard.ai/knowledge/how-to-test-ml-models-4-metamorphic-testing.

Evaluating HMT, including appropriate operator trust in the AI, will also need to be included in test planning and conducted during detailed design and development (Pham 2022; Tate 2021). One potential framework for planning and conducting such testing is provided in the Systems Engineering journal article, “A Team-Centric Metric Framework for Testing and Evaluation of Human-Machine Teams” (Wilkins et al. 2024), which emphasizes the need for consideration and evaluation of metrics for multiple aspects of the interactions among and between the HMT. See Section 2.2.3.4 for specifics.

Test planning should also consider the information needed to obtain all needed certifications, including safety, interoperability, and cybersecurity. These certifications are required before fielding and before OT. (See Sections 2.2.2.2, 2.2.2.3, and 4.6. Additional information will also be provided in the forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook.) The forthcoming DTE&A guidebook discussing T&E of M&S as well as the VV&A Recommended Practices Guide Website will contain material generally applicable to any M&S used by DoD.

Post-Fielding

T&E planning is necessary to ensure that the AIES/MLES continues to perform as intended after it is fielded. Changes in ML model performance (e.g., drift (gradual or sudden), emergent behavior, degradation in robustness) due to inputs encountered during operations must be detected and evaluated (Chandrasekaran et al. 2023). When purposeful changes are made to the ML model, the TVT datasets as well as performance metrics for overall system performance may need to be updated. Depending upon the nature of any changes made, the data and revised model may also need to undergo VV&A again. In other words, the need for planning to iteratively and continually move through the ML pipeline post-fielding and to provide recertification of compliance with DoDD 3000.09 requirements should be considered, as should the need to reevaluate overall system performance. As a result, test teams, tools, and infrastructure that might have been substantially reduced, eliminated, or closed during the post-fielding phase of a non-AIES/MLES will need to be maintained with appropriate resourcing.

Planning support for all the phases discussed in Section 2.4.4.1 must account for the government’s need for, timely access to, and involvement in T&E information and activities to support the government’s evaluations of AIES/MLES, including assessments of program progress and resource needs. Contractual provisions will be necessary to ensure this access and involvement.

2.4.5 Test Preparation and Execution

Early Development

The early development activities associated with collecting and preparing TVT data will likely involve substantial manual effort even with automated assistance (e.g., tasks such as identifying and obtaining data; mitigating problems arising from missing or incomplete records, outliers or anomalies, improper formatting or structuring, limited or sparse features orattributes; and determining the need for feature engineering; DataRobot 2024). Nonetheless, the use of automated tools can be helpful for data preparation, as well as for several other aspects of development and T&E (for more information see the T&E Strategy Frameworks on the CDAO JATIC Documentation Website). Consequently, preparation and execution of independent government T&E of these data to evaluate whether their preparation has been accomplished properly may also require substantial manual effort, as well as close consultation with the data scientists who performed the preparations. To the extent reporting by those data scientists will be used for evaluation, the content and frequency of those reports will need to be defined contractually.

The activities for VV&A of the TVT data to evaluate whether the similarity and dissimilarity among them is appropriate may be largely automated, given the up-front preparation of needed coding or scripts for ingesting the datasets and computing metrics. Evaluation of whether the data are consistent with the likely operating environment and CONEMP will likely require manual review of the data with automated assistance. System users and operators will need to be fully involved in this step to avoid late discovery (during detailed design or fielding) of significant inconsistencies that would likely be expensive to mitigate. Provisions will need to be defined contractually for the involvement of government users and government testers in executing the T&E.

Test preparation and execution will likely be accomplished continually and iteratively and will be integral to the work of early development. The data scientists preparing the data will also be heavily involved in, if not solely responsible for, the data’s T&E and VV&A activities. Independent T&E by government testers will need to be accomplished in close consultation with those data scientists. Reporting by the developers or involvement by the government in execution of the T&E will need to be defined contractually.

Detailed Design and Development

Preparation for and execution of T&E of the stand-alone model will be an integral part of the model’s training. Training will be iterative with feedback based on the metrics and techniques, described in Sectio 2.4.4, used to modify the model’s structure and parameters until satisfactory results are obtained. Preparation and execution will be mostly automated. Information supporting

VV&A of the stand-alone model will likely be collected as part of the model's training. Contractual provisions will need to be made for government access to this information and for any associated specific or unique government reporting or involvement needed to satisfy the AI-/ML-specific requirements of DoDD 3000.09 or DoDI 5000.61.

Integration of the ML model with the overall system will likely proceed in stages. For example, the model could first be integrated within purely digital simulations of the system, followed by combined hardware-in-the-loop and digital environments, then in prototype hardware and software, and finally in pre-production hardware and software. As the realism of the environments within which the ML model is used increases, so too will the manual activities, complexity, and time needed to prepare for and execute T&E. To the extent problems are discovered that are traceable to the ML model, the steps composing the ML pipeline may have to be repeated in whole or in part. If so, test planning, preparation, and execution will become increasingly complex as the need to iteratively repeat, in whole or in part, the T&E and VV&A of the ML model conflicts with the desire to continue detailed design and development of the overall integrated system.

Post-Fielding

Preparation for and execution of T&E will be analogous to that discussed for detailed design and development. Complexities will depend upon the extent to which only the stand-alone ML model or the overall integrated system is undergoing T&E, as well as the extent to which VV&A of the model must be reconducted or compliance with the requirements of DoDD 3000.09 must be recertified.

2.4.6 Test Analysis and Evaluation

T&E analysis and evaluation will be conducted mostly in parallel across phases to support iterative design and development of both the ML models and the overall integrated systems. However, exceptions may exist for analyses to support the preparation of independent evaluations for milestones and other decisions by government offices (see Section 2.2.4.2). These exceptions may require special-purpose analyses of test data not otherwise conducted during development (e.g., assessing compliance with DoDD 3000.09). In general, throughout the system life cycle, test data from the entire development history will be relevant to evaluations and should be retained for that purpose.

Early Development

In early development, timely analysis and evaluation of test data on the adequacy of datasets and attempts at training prototype alternative ML models will be needed to support decisions on approaches to be carried forward into detailed design and development.

Detailed Design and Development

Timely analysis and evaluation of test data on dataset adequacy and prototype ML model performance will be needed in the earliest part of this phase to support the development of the RFP, the decision to release the RFP, and the evaluation of the RFP responses. As is the case for any program, as detailed design and development proceeds, analysis and evaluation will be conducted along with execution to identify issues and problems, solutions, necessary programmatic changes, and overall progress toward achieving a fieldable system. Toward the end of this phase, comprehensive evaluations of test data supporting decisions on proceeding with OT (see Section 2.2.4.3), production, and fielding will be needed. Throughout this phase, AI-/ML-specific issues regarding performance and risk must be addressed, especially as the phase concludes (see Sections 2.2.1 and 2.2.2). Cogent analyses will be needed to assure decision makers that the AIES/MLES will function as intended, is robust, and complies with the requirements of DoDD 3000.09. These requirements will be formally assessed during a pre-fielding review.

Post-Fielding

Analysis and evaluation will be constructed along with testing to identify and correct issues such as drift in performance and emergent behavior of the AI/MLES. The evaluations will need to provide the information needed to support decision making on whether the system's performance remains consistent with continued operational use, as well as with the requirements of DoDD 3000.09. If it does not, further analysis and evaluation will be needed to assess the corrective actions and resources required to ensure the system's continued effectiveness.

2.5 Summary

This section examined the recent advances in AI systems that have implications for DT&E responsibilities in performance evaluation, risk assessment, and support to systems engineering. The issues raised by AI, and in particular by ML, were introduced at a high level. Section 3 presents specific T&E methodologies relevant to the novel challenges of ML. Section 4 addresses expanded organizational and professional interactions outside the T&E career field.

This section concludes with a very high-level view of "what's new" about the implications for T&E discussed throughout this section above: The driving feature in the testing of AI is that comprehensive testing is no longer feasible for many AI components or AIES. Comprehensive testing for systems with a large state space can occur only when it is possible to describe the performance envelope in which the system will operate; interpolate between test points; and extrapolate test results beyond the tested region. The lack of predictive underlying theory for ML systems makes comprehensive testing no longer feasible. In addition, ML systems often exhibit unpredictable behavior with small changes in input, further complicating the use of comprehensive testing. Finally, the CONEMP for MLES calls for continuous retraining and additional learning, meaning that fixed test configurations are a thing of the past. Under these conditions, the traditional test design and strategy are insufficient.

This absence of fixed test configurations changes the relationship between *testing* (i.e., measurements) and *evaluation* (i.e., assessments). Although the need for evaluation remains, this evaluation must have applicability beyond the executed test conditions and even beyond the tested version(s) of the ML model. This relationship change introduces a new class of challenges in generalizing results beyond the specific conditions of the performed tests. Characterizing the future capabilities, limitations, and risks of the AIES is now an undertaking with increased scope. A wider set of evidence-generating activities and analyses may be needed to support arguments that establish the desired dependability and inform stakeholders regarding the likely consequences of fielding and employment.

This increased scope of evaluation will likely increase both cost and schedule impacts. Although specific cost and schedule estimation techniques for AIES development are beyond the scope of this guidebook, DT&E personnel should be aware of the need for additional testing and evaluation iterations.

3 AI-Driven Changes in T&E Practice

3.1 Introduction

Section 2 provided an overview of areas in which the use of AI might affect how DT&E is planned and executed, either for complete systems or during development. It also introduced, at a high level, specific T&E activities, approaches, and techniques that might be useful in dealing with these AI-induced changes. This section provides additional details on how and when to implement several of these activities, approaches, and techniques. Subsections focus on data and model quality, critical to all ML applications; the use of formal methods to supplement empirical testing; how the use of AI interacts with M&S; techniques enabling visibility into ML models; and how the impacts of AI on T&E are distributed across the acquisition life cycle, from early involvement through fielding and beyond. This section closes with discussions of how evaluation, reporting, and support to certification are affected.

This guidebook describes a comprehensive approach to conducting T&E of AIES. Table 3-1 provides a step-by-step summary of that approach, including a comprehensive list of the questions it prompts and helps answer. Not every question will need to be addressed for each program. The questions that are most relevant will depend upon the program's specific content, mission, and time-phasing within the acquisition life cycle.

Table 3-1: Crosswalk of T&E Questions and Guidebook Sections

Step or Question	Guidebook Section (with cross-reference link)
Step 1: Evaluate and conduct VV&A of the data.	
Do the data cover the system's operational domain?	2.2.1.2; 2.2.3.1; 2.2.3.3; 2.4.5; 3.2.2.3
Do the data's features compose a good statistical representation of the operational domain?	2.2.3.1; 2.2.3.3; 3.2.2.3
Are there formal methods that can be used to verify that the data have been obtained/extracted properly?	3.4.5
Have the data been properly transformed and normalized?	2.2.3.1; 2.2.3.3; 3.2.2.1
If augmented using synthetic data, do those data have the proper features and are they representative of the operational domain?	2.2.3.1; 2.2.3.3; 3.2.2.1; 3.2.2.2
Have the data been evaluated for poisoning?	2.2.3.3; 2.3.3; 2.3.4; 3.2.2.3
Are there sufficient data for them to be partitioned into statistically representative training, validation, and test sets?	2.2.3.3; 3.2.2.1
Is the partitioning scheme used to determine the training, validation, and test datasets defensible?	2.2.3.3; 3.2.2.1
Have provisions, including contractual provisions, been made for access to all the data used to train, validate, and test the ML models?	2.2.4.4; 2.4.4.1; 2.4.5; 3.3; 4.2

Step or Question	Guidebook Section (with cross-reference link)
Step 2: Evaluate model quality and conduct VV&A of the ML model.	
Did the model's convergence to its final hyperparameters (e.g., optimization heuristic, number of neural network layers of number of clusters) proceed without substantial oscillation?	3.2.3; 3.2.4
Did the model's performance using the test dataset differ substantially from its performance using the training and validation datasets?	3.2.3; 3.2.6; 3.5.1
If regression was used for optimization, did the method appropriately balance worst-case performance and average performance? Did it mitigate the potential for overfitting?	2.2.3.2; 2.4.4.1; 3.2.4.2
For ML classifiers, are the model's accuracy, precision, and recall consistent with its mission/use?	2.4.4.1; 3.2.4.3
For clustering models, is the distance metric used to measure similarity among the data consistent with the intended mission/use? • Does adding a small amount of random noise to the input cause substantial changes in either the number or composition of clusters? • Do transformations of the data (such as converting categorical data to numerical scales) cause substantial changes in either the number or composition of clusters? • Do erroneous data points or other extreme outliers affect clustering?	3.2.4.3
Have input-based explainable AI strategies (e.g., drop column, saliency map, Shapley additive values, permutation and local interpretable model-agnostic explanations) and/or metamorphic testing been used to evaluate model behavior?	3.5.3.1
For reinforcement learning, is the reward function consistent with the ML algorithm (model-, policy-, or value-based) and with the intended mission/use?	2.1.2; 3.2.4.1; 3.2.4.5
Can formal methods (e.g., static analysis, model checking, deductive verification, proof assistants) be applied to evaluate the model?	3.4.1
Do data and model cards exist providing traceability (i.e., the ability to reconstruct how an ML algorithm came to behave the way it does)?	2.2.2.1 (see Traceable AI-Enabled Systems); 2.2.3.3; 3.5.4
Have potentially hazardous states of the ML model been identified for later evaluation supporting evaluation of overall system safety?	3.4.2
Have brittleness and robustness been adequately characterized?	2.2.1.1; 2.2.2.2; 2.2.2.3; 2.4.1; 2.4.3; 2.4.4
Step 3: Characterize performance of the system with AI embedded.	
Have the data needed to support safety documentation been identified? Are formal methods available (e.g., Satisfiability Modulo Theory, Mixed-Integer Linear Programming, Global Optimization) to demonstrate that potentially hazardous states will not occur?	2.2.4.3 (see AI-/ML-Specific Operational Test Readiness Review Considerations); 2.4.2; 2.4.4.1; 3.4; 3.5.2

Step or Question	Guidebook Section (with cross-reference link)
Have system-level performance metrics been defined with participation by the system's users?	2.3.3; 2.4.4
Is the CONEMP well-documented and are there plans in place to evaluate performance across its full extent and with sufficient density of test points using live testing and M&S to detect anomalous behavior? • Have plans for T&E incorporated the need to be able to accurately characterize the differences between average and worst-case behavior? • Have plans for testing incorporated obtaining the information needed to evaluate human-machine interactions/teaming? • Have plans for T&E incorporated obtaining information to evaluate the system's interoperability with that of other systems and to provide reasonable assurance that AI interactions with other systems will cause no harm? • Have plans for cybersecurity T&E been made consistent with evaluating adversarial actions/effects on the system and performance against other threats consistent with the forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook?	2.2.2.1 (see Governable AI-Enabled Systems); 2.2.3; 2.4.2; 3.6
Have real-time diagnostics been included in the system to provide the adequate information enabling operators to understand and trust performance; i.e.: • When employed correctly, the system will dependably do well what it is designed to do? • When employed correctly, the system will dependably not do undesirable things? • When paired with the humans it is intended to work with, the system will dependably be employed correctly?	3.5
Has runtime monitoring been implemented and are plans in place to test and evaluate its ability to detect and prevent undesired system behavior?	2.2.2.2; 2.4.4.1; 3.5; 4.4.2; 4.5.1.2
What live test infrastructure exists (or should be developed) enabling safety data and performance metrics to be collected under both controlled and more realistic mission-representative conditions? What steps need to be taken to use that infrastructure?	2.2.4.2
Does M&S having undergone VV&A exist that can be used to evaluate the performance of the overall system? • Does the M&S incorporate realistic sensor inputs including all key features of the operational domain? • Does the M&S incorporate accurate timing of sensor and ML model inputs and outputs, as well as hardware response times? • Does the M&S incorporate important features of the real environment, such as the stochasticity inherent in the sensor environmental inputs, actuator responses, and computational timing?	3.3

Step or Question	Guidebook Section (with cross-reference link)
• Have the appropriate statistical methods been employed to compare and evaluate the performance predictions of the M&S with data for the performance of the actual system?	
Step 4: Track performance of the fielded system.	
Are plans and processes in place to track model and data/concept drift?	3.2.2.3; 3.2.6
Will the runtime diagnostics, M&S, and test infrastructure used to evaluate system performance during DT&E be resourced and available to track performance of the fielded system?	2.2.4.2; 3.3; 3.5
Have metrics been developed consistent with the available test infrastructure and diagnostics enabling decisions to be made regarding the need for retraining and/or updating the ML models to correct model and concept drift?	3.2.6
Will the ability to obtain and prepare data, train the ML models, and evaluate their performance, including both personnel and infrastructure, be available and resourced? Will the personnel and infrastructure used to perform steps 1 through 3 be sufficiently preserved to enable model retraining/updating and evaluation?	See previous Section crosswalks

3.2 ML Model Development and Assessment

Although many kinds of ML models exist, they all share a common high-level structure and development process with four main phases:

1. Model Specification
2. Data Preparation
3. Model Training
4. Model Assessment

3.2.1 Model Specification

In the model specification stage, the developers (provisionally) choose what kind of data the model will operate on; what outputs it will produce; what functional form the model will use; and how success will be measured. Most ML models' functional form will be a parametric model: a functional form with many unknown coefficients, capable of fitting to a wide range of data patterns. In the ML community, specific families of parametric models are sometimes called "architectures," not to be confused with software architectures in the more general sense. Common SL architectures include neural networks, decision trees, Bayesian belief networks, and support vector machines. UL architectures include k-means clustering and kernel estimation models. RL often uses Markov decision process models. GenAI applications often involve so-called transformer models or long short-term memory models (e.g., Badillo et al. 2020).

To measure how well the model succeeds at its assigned task during training, the developers will choose an objective function: a numerical measure of goodness, intended to quantify how well the current set of parameter values achieves the goals of the ML model. The ML's learning algorithm attempts to optimize the objective function by setting the coefficients. Because the objective function is the only guidance the learning algorithm has, in terms of what desirable outputs look like, it is very important that testers know what objective function was selected and verify that the objective function training goals are well aligned with actual operational goals and trade-offs in the mission environment.

3.2.2 Data Preparation

SL, UL, and GenAI models depend on datasets from which to learn. These datasets are used for model training, validating, and testing, as described in Section 2.4.1. The appropriateness of the TVT dataset to the intended mission and environment is critical for ML success. T&E has a role in assessing the adequacy of the data for the intended purposes.

The nature of the instances in the TVT dataset depends on which category of ML is being developed. For example, the dataset for SL consists of instances of the input type chosen for the ML model to operate on, together with each instance's associated label(s). For unsupervised clustering applications, the TVT data consist of unlabeled data instances for which a distance metric has been defined. Thus, the category of ML will influence the T&E role in assessing the adequacy of the data.

In general, raw data are poorly suited to train ML models. The ML data preparation process typically involves some or all of the steps shown in Figure 3-1 and discussed below.

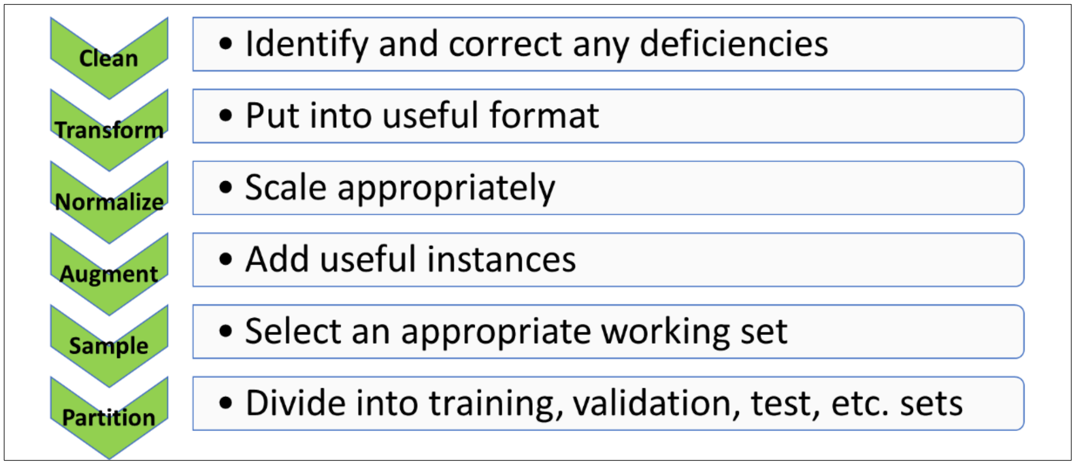

Clean
• Identify and correct any deficiencies
Transform
• Put into useful format
Normalize
• Scale appropriately
Augment
• Add useful instances
Sample
• Select an appropriate working set
Partition
• Divide into training, validation, test, etc. sets

Figure 3-1. ML Data Preparation Process

Clean: Cleaning data may include identifying and correcting (or removing) erroneous data; fixing incorrect data labels; eliminating outliers in numerical data; and disguising any personally identifiable or sensitive data.

Transform: Transforming the data may include aggregating certain numerical data into categories (e.g., coding ages into categories rather than numeric ages or coding zip codes into categorical regions). Transformation can also go the other direction, converting categorical data to numerical scales or combining or regrouping categories to reflect attributes relevant to the specific application.

Normalize: Normalizing data may involve converting data to consistent units; applying transformations to numerical data where appropriate (e.g., taking logarithms of data that range over several orders of magnitude); or putting all input ranges on comparable scales.

Augment: Augmentation of the data may include imputing missing values, which is usually done by estimating the missing fields of incomplete data instances using statistical averages over the complete instances. Another form of augmentation is creating synthetic data. Synthetic data are discussed in more detail in Section 3.2.2.2.

Partition: Partitioning of data is the separation of the TVT data into a combined T&V set; a set-aside test set used to assess how well the model has learned; and additional set-aside independent test sets as needed. Further details on data partitioning are discussed in Section 3.2.2.1.

Ideally, DT&E personnel should be aware of how the raw data were cleaned, transformed, normalized, augmented, and partitioned. Use of rich data cards and model cards (see Sections 2.2.2 and 2.2.2.3) can support this goal. Awareness of the steps used in the data preparation pipeline can alert testers to specific issues to watch for. For example, if the data required a great deal of missing value imputation or other augmentation, it is important to verify that the resulting TVT set is still representative of the operational environment. Similarly, testers may wish to verify that the data partitions reserved for test and independent test are at least as diverse and complete as the T&V partition.

3.2.2.1 Data Partitioning

As described in Section 2.4.1, the validation step in training an ML algorithm involves assessing the model's performance on data instances that were not used in the training. In addition, evaluating a trained model requires not merely performing a final validation but also testing the model on data that were never used at any point to either train or validate the model. This gives insight on the model's generalizability—how well it will perform on never-seen-before inputs. Testing on separate test data can also help determine whether the model is biased or overfitted to its training data (for SL).

A common practice is to set aside a portion of the training dataset (typically 10 to 30 percent) as testing data, which will not be used for training or validation during learning. The testing data may be further subdivided into a test dataset to be used by the developers and one or more independent test datasets for use by external T&E organizations; data cards may be useful here to ensure traceability of the data. After setting aside the test datasets, the remaining data make up the T&V dataset. It is important to verify that the partitions are statistically similar and sufficient in size, coverage, and representativeness.

Training an ML model is an iterative process; at each iteration, the learning algorithm uses information from previous iterations and heuristic optimization to find a set of coefficients for the parametric model that perform better on the training dataset. In each iteration, the T&V dataset is partitioned into training data that the model will learn from and validation data that will be used to assess learning and guide the learning algorithm. A common way to split the T&V dataset is *k-fold cross-validation*, in which the T&V data are partitioned into k subgroups (folds), and the model is trained on (k-1) folds and validated against the remaining fold. This process is repeated k times, so that each permutation of training and holdout set is applied.

This entire process is repeated until the model parameters converge. Once there is no more improvement and a model has been selected, the full T&V dataset is used to train the model, and its performance is evaluated by the test dataset.

3.2.2.2 Synthetic Data

Synthetic data are any TVT instances that are not directly derived (e.g., by transformation and/or normalization from real-world data). Many techniques exist for generating synthetic data, some of which are extremely sophisticated. Synthetic data might include instances created by perturbing or manipulating real-world instances or instances created by statistically averaging or otherwise blending more than one real-world instance. Synthetic data could also be created as outputs of accredited M&S or by imputing missing values in real-world data. Some training methods, such as generative adversarial networks, generate synthetic validation data as part of the learning algorithm (e.g., Nikolenko 2021).

The use of synthetic data to train ML poses several novel challenges for DT&E, including the following:

- Verifying generalizability. Because ML algorithms typically learn high-dimensional correlations in the training data and use those correlations to make predictions, models learned from synthetic data will only generalize correctly in real-world operations if the synthetic data share the relevant correlations with real-world data. Typically, neither developers nor operators know in advance which relationships in the data will turn out to have the most predictive power. The best they can do is attempt to verify that the synthetic data and the real data are statistically similar, and then watch closely during the learning process for signs of divergence between performance on real inputs and performance on synthetic inputs. Principal components analysis (PCA) is one useful method for comparing the distribution of the synthetic data to that of the real data. If the results of PCA applied to just the synthetic data yield a very different result from PCA applied to just the real data, that suggests a flaw in the synthetic data generation process that may hinder generalizability.

- Establishing traceability. RAI policy requires that AI used in defense applications be traceable. Provenance of TVT data can be complicated even in the absence of synthetic data if significant data normalization or transformations were applied. The use of synthetic data adds an additional layer of separation between reality and the trained model. It may be helpful to apply XAI techniques separately to characterize both the model behavior on real-world inputs and the model behavior on synthetic inputs to detect any differences in which features of the data are most influential. This not only can identify potential issues in synthetic data generation but also can be used to supplement the validation step during learning.
- Accrediting synthetic data for specific use cases. As discussed in Section 3.2.2.3, DT&E activities can support required VV&A of ML models and the data used to develop them. The generalizability issues associated with synthetic data make accreditation of synthetic TVT data particularly challenging. It is already difficult to judge whether synthetic TVT data adequately support the specific mission in question, much less which other missions and environments they might also adequately support. In addition, testers should be aware that VV&A of training datasets that include synthetic data might require understanding the methods that were used to generate those synthetic data and that M&S used to generate synthetic instances might require additional accreditation of both the M&S model and the data for that use. See Section 3.2.2.3 for more information on data accreditation.

3.2.2.3 Verification, Validation, and Accreditation of the Dataset

DoDI 5000.61 leaves the details of VV&A on data and models to the DoD Components. It is anticipated, however, that the T&E community will insist on being the accreditation authority for any data partitioned for independent test. As part of TEMP development and T&E Working Group participation, T&E practitioners will need to ensure sufficient access to partitioned data—and in some circumstances all the training data—to address the data issues discussed above.

In accordance with DoDI 5000.61, models and associated data used to support DoD processes, products, and decisions must undergo V&V throughout their life cycles and must be accredited for a specific intended use.

ML can be influenced by various data problems; thus, T&E can play an important role in the VV&A of data to identify issues such as erroneous data (corrupt/spurious data, data poisoning, mislabeled data, missing fields); unrepresentative data (overrepresented classes/giraffing,[10] underrepresented classes, synthetic data quality); and insufficient data for learning (for a specific class or the overall dataset). VV&A may also help identify improper scaling/normalization; privacy issues; security issues; timeliness/concept drift; or unwanted bias. In terms of the development pipeline discussed in Section 3.2.2, verification activities would be closely linked to the "clean" and "normalize" steps. Validation and accreditation depend upon the "sample" step (see Figure 3-1); an "appropriate working set" is one that has not only quality data but also data that are sufficient for addressing user needs with respect to environment and mission (validation) and that do so well enough for the particular application (accreditation).

DT&E activities may be used to support VV&A of the data used to build the ML model. This role extends across the ML life cycle, from pre-development verification (including any synthetic data) to post-fielding accreditation for new mission areas.

In the context of an MLES, verification answers this question: Are these data suitable for training the ML model? Verification establishes that the TVT dataset is correct, well formatted, normalized, and timely and contains a sufficient number of instances to support training, validation, test, and independent test. Verification also ensures that the data are free of malicious actions (such as data poisoning) and unwanted bias. Exploratory data analysis is a blanket term for a collection of visualization and statistical data exploration techniques (box plots, histograms, cumulative distribution functions, etc.) that can be applied to large, high-dimensional datasets. These techniques can help uncover various data issues such as unrepresentativeness, outliers and anomalies, redundancy, or formatting errors.

Validation goes beyond the ability to train a model; it looks at whether the data support training the correct model for the purpose, and it answers this question: Are these data mission appropriate for *this* intended use? Validation is concerned with not only the correctness of the data but also the coverage of realistic mission inputs; whether labels include all mission-relevant categories; and the extent to which data are representative with respect to frequency and co-occurrence of categories and situations. Validation is also used to determine whether the data are secure and privacy issues are mitigated.

[10] The case when an ML model identifies objects such as giraffes in pictures when none exist because those objects (e.g., giraffes) are overrepresented in the training dataset, and pictures without the objects are underrepresented.

Accreditation goes beyond validation to ask the following question: For which tests and missions could this dataset be used to train an appropriate model?—that is, for which other ML purposes is the dataset valid. Accreditation considers the operational environments and conditions; the range of mission goals; human-machine CONEMPs; and associated simulation models. Other questions of interest related to accreditation include the following:

- Can we extrapolate what the consequences might be of using these data for purposes outside their accredited uses?
- Does performance degrade gracefully at the boundaries, or can it suddenly break?
- Can we map out where performance changes are abrupt or smooth?
- What additional T&E would be needed to validate the dataset for a new proposed test or mission context outside its current accreditation?

ML's dependency on data can introduce new risks (malicious actions or flawed data) as well as make it more difficult to evaluate system technical performance. DT&E is a natural contributor to VV&A of the data that will be used to build ML models. This role extends across the ML life cycle, from pre-development verification (including evaluation of any synthetic data) to post-fielding accreditation of datasets for use in new mission areas. As of the time of writing, no official guidance on how to comply with the data VV&A requirements of DoDI 5000.61 has been issued.

Data partitioning, the use of synthetic data, and the VV&A of the data all have implications for DT&E. The quality of the data partitioning must be understood; a best practice is having T&E professionals participate in the partitioning. Determining that synthetic data adequately represent the real-world environment can require additional testing, beyond what is needed for "physical" data. T&E professionals should be engaged in V&V activities directly and, under some circumstances, may be accrediting authorities.

3.2.3 Model Training

Training an ML model is an iterative process that uses a learning algorithm to find a parametric model that optimizes the objective function for the given T&V dataset, as discussed in Section 3.2.2.1. Based on the results, the learning algorithm's hyperparameters that define the model and control the optimization can be adjusted. Hyperparameters include the choice of optimization heuristic; the convergence rates and tolerances of the heuristic; and the structural features of the model (number of layers in a neural network model or number of clusters in a clustering model).

Although DT&E is not normally involved in the execution of model training, it may be important for testers to be aware of exactly what model performance metric was optimized during the training process. This may require knowledge of the objective function and hyperparameters, which are discussed in more detail below.

3.2.4 Model Assessment

Once performance on the validation dataset is no longer improving, the learning phase is complete, and the hyperparameters and data representation are fixed at their current values. At this point, the now optimized learning algorithm is usually applied one last time to the entire T&V dataset for maximum information extraction. The resulting final trained model is then assessed using the reserved test data, and performance measures are computed. Performance measures are discussed later in this section.

The choice of data, parametric model, and learning algorithm will drive the quality of the model, and interplay occurs between each of these areas. Although certain combinations of architecture, algorithm, and data might not exist presently, every ML model involves a combination of these features. These are the drivers of model quality, but model quality makes sense only in the context of the model's mission set—quality does not exist in a vacuum. Just because an ML model generalizes the training data well, it does not mean the model is "good." Many other dimensions of system performance (adherence to RAI policy, safety, reliability, etc.) might be relevant to operational utility. In general, these other dimensions can be assessed only with respect to a specific intended mission and operational environment.

In some cases, DoD systems may involve components or platforms that are ML enabled but where the government does not have the ability to query the ML model in isolation, much less examine its internal structure. Government DT&E will need to characterize system performance, limitations, and risks without the benefit of being able to directly assess the ML model's performance, even using black box methods. The test strategy in such cases should include tests designed to detect the specific failure modes that ML is prone to—namely, sensitive dependency to small changes in input; poor worst-case performance; emergent behavior; unwanted bias; etc. Use of M&S may be particularly important in these cases to conduct red teaming and other adversarial test strategies to detect and characterize worst-case and emergent behaviors. The remainder of this section assumes that testers have, at a minimum, the ability to feed chosen inputs to the ML model in question and determine the corresponding output.

3.2.4.1 Assessing ML Models

Each kind of ML has its own appropriate metrics for model performance. This section describes some of these metrics. Because metrics are used during the training process to assess model performance, it is an important role of DT&E to verify that the metrics used to assess model quality during training align appropriately with mission-level performance goals and measures of effectiveness. The section concludes with some notes of DT&E support to VV&A of ML models and the problem of model and data drift.

3.2.4.2 Testing and Metrics for Different ML Categories

The type of metrics used to determine the average behavior will depend on the category of the model. This section considers the three main ML categories: SL, UL, and RL.

Models with supervised algorithms usually segment data into two to three groups: T&V, testing, and (sometimes) development. Models that use unsupervised algorithms usually do not segment data in a meaningful way—the desire is pattern recognition and insight. Models with reinforcement algorithms generate data on the fly or use simulation (see Section 3.3). Other types of algorithms are variations on these three themes. A T&E expert should ensure that the model developers clearly lay out the following: (1) the metrics used to train and (if different) evaluate the model; (2) the data segmentation schema (how the TVT data were partitioned) used for the model; and (3) any other performance measures that are not used in model training and evaluation but are operationally relevant. Program officials and SMEs should verify that the metrics used to train and evaluate the model align properly with mission goals and constraints. Using the wrong metric to evaluate model performance can be more harmful than using no metric at all.

3.2.4.3 ML Regression Metrics

Regression involves selecting the parameters of a function that predict the value of a numeric output variable based on given values of a set of input variables; the goal of a regression analysis is to capture the trends of hypothesized linked variables within the data but not the noise. Regression metrics are a function of the size of the prediction errors—the residuals—over the test dataset.

Commonly used regression metrics include traditional summary statistics such as MSE, MAE, and the coefficient of determination (R^2). More sophisticated ML regression performance metrics include cross-entropy loss (Mao et al. 2023) and minimax regret (Savage 1951). These are functions of the residuals and vary mainly in how much they penalize errors of different magnitude. For example, MSE penalizes errors quadratically so larger errors are penalized more severely than smaller errors, whereas MAE penalizes errors linearly. Minimax regret ignores average error and focuses on worst-case differences between the output value and the correct value. Classical statistics tend to use MSE, partly for theoretical reasons and partly for computational convenience. This is equivalent to minimizing the variance (or standard deviation) of the errors.

In ordinary least squares regression, there is an assumption that residuals are normally distributed; this is not true, however, for many ML regression models. In practice, ML errors do not exhibit smooth Gaussian tails. Extreme outliers are not unusual, especially for inputs far from the center of the TVT data, and small changes in the input can cause large changes in the output. These behaviors may lead to unacceptable worst-case system performance.

It is possible to adjust the learning algorithm's objective function to get better worst-case performance, at the expense of worse average performance. For example, minimizing the maximum absolute residual provides a bound on how robust the model can be made, given the available training/validation/test data. Other techniques, such as lasso—least absolute shrinkage and selection operator—regression (Tibshirani 1996) and ridge regression (McDonald 2009), impose an extra penalty on nonzero coefficients to help avoid overfitting. In general, methods that trade average or best-case performance to improve worst-case performance are called "robust machine learning" (Guerraoui et al. 2024).

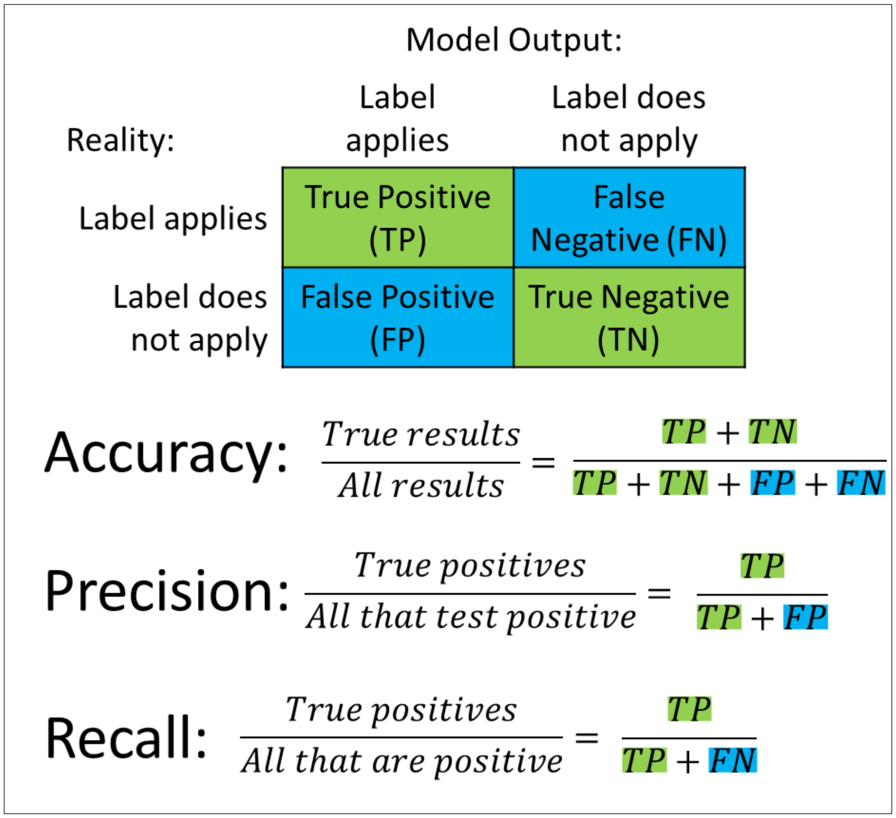
The traditional regression metrics are useful, but the output of ML models frequently differ substantially from the assumptions behind the metrics. Caution—and potentially additional measurements exploring outliers—may be called for.

3.2.4.4 Classification Metrics

For classification models, four possible outcomes exist when testing whether a given label applies to an input:

- True positive (TP): The label applies, and the model says that it applies.
- False positive (FP): The label does not apply, but the model says that it does apply.
- True negative (TN): The label does not apply, and the model says it does not apply.
- False negative (FN): The label applies, but the model says that it does not apply.

These outcomes are typically all that will be measured. The evaluation must tie these outcomes to mission effectiveness, which will depend upon the specifics of what the model got right and what it got wrong. Several standard metrics are used for this. The first and simplest metric is accuracy: the fraction of outputs that was correct. Accuracy, however, is not always enough to measure operational usefulness. A terrorism prediction model that always says "terrorists are not about to attack" will be 99.99999 percent accurate but useless for its intended purpose. Thus, other metrics should be considered. Precision measures the reliability of positive outputs. It determines the likelihood that an instance labeled as positive really is positive. Recall is another metric (also known as sensitivity or P_d) that shows what fraction of positive instances was successfully identified. Figure 3-2 shows how the three metrics—accuracy, precision, and recall—are calculated.



Reality:	Model Output: Label applies	Model Output: Label does not apply
Label applies	True Positive (TP)	False Negative (FN)
Label does not apply	False Positive (FP)	True Negative (TN)

$$\text{Accuracy:}\quad \frac{\mathit{True\ results}}{\mathit{All\ results}} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision:}\quad \frac{\mathit{True\ positives}}{\mathit{All\ that\ test\ positive}} = \frac{TP}{TP + FP}$$

$$\text{Recall:}\quad \frac{\mathit{True\ positives}}{\mathit{All\ that\ are\ positive}} = \frac{TP}{TP + FN}$$

Figure 3-2. The Confusion Matrix, Illustrating Classification Algorithm Performance

Note that whereas accuracy treats positive and negative instances equally, both precision and recall are focused on the model's behavior regarding positive instances. Both metrics ignore TNs, which can be a major driver of accuracy. If TNs are important to mission success, both precision and recall may be misleading metrics for model performance. Additionally, several commonly used metrics attempt to combine precision and recall into a single score that reflects the importance of both; for example, *F_1 score* is the harmonic mean of precision and recall, and the *Fowlkes-Mallows (F-M) index* is the geometric mean of precision and recall. Both F_1 score and F-M index treat precision and recall as equally important and identically scaled. For most real-life missions, it is not true that decision makers are indifferent among models with the same F_1 or F-M score. Some levels of precision or recall are unacceptable. For a given system, an increase in recall might be valued more than a comparable increase in precision, or vice versa. An important role of T&E is in verifying that the relative importance of precision and recall to the warfighter is reflected in the objective function being used to train the ML model and in the metrics used to assess model performance. This will generally require interaction with warfighters and, potentially, other communities.

Ideally, classification models should have both high precision and high recall. If that is not possible, it is necessary to trade between them. This can be done either by adjusting the objective function's decision threshold used during training or by using a different ML model. In general, modifying a classification model to increase the probability of recognizing a given class will also increase FPs for that class. The classic version of the ROC curve can be used to plot the TP rate on the y-axis and the FP rate on the x-axis to visualize the trade-off. It is also common to see similar plots of precision versus recall. ROC curves can represent either how performance changes as a function of the penalties used in training or how different types of models behave for a given problem.

The metrics discussed above are for a binary classification task; however, these metrics can easily be extended to multiclass classification. For multiple classes, accuracy is measured the same as in the binary case—that is, the ratio of correct predictions to all predictions across the four classes. For precision and recall, two methods are used to calculate these metrics: macro-average and micro-average. The macro-average precision/recall is the sum individual precision/recall of each class divided by the number of classes. The micro-average is the sum of TPs of each class divided by the metric's respective denominator (sum of TPs and FPs of each class for precision; sum of TPs and FNs of each class for recall; Grandini et al. 2020).

One other important difference in multiclass classification is the potential of hierarchical classes. For example, in classifying animals, there could be one parent node for the primate class that could have local nodes for chimp, gorilla, and human subclasses.

3.2.4.5 Unsupervised Learning – Clustering Metrics

Clustering—grouping inputs into subsets whose members are related or similar—is the most common form of UL currently in use and arguably the most relevant to defense applications. The purpose of clustering algorithms is to assign every instance in the dataset to a group to the extent possible so that every group member is (in some sense) like every other member of that group but unlike every member of the other groups. Clustering model development always begins with specifying a distance metric on instances that quantifies "similarity." The clustering algorithm attempts to minimize the within-cluster distances while maximizing the distances between clusters. This distance metric is intended to be highly correlated with the kind of similarity important for the mission. The distance metric is derived from the data elements of the instances (height, weight, pixel values, word frequencies, etc.). If the distance between two elements is zero, then they will always be assigned to the same cluster, regardless of other data.

The two ways to evaluate clustering are internal evaluation and external evaluation. Internal evaluation metrics use only the unlabeled data and the results of the clustering to measure the performance of the model. Many different internal evaluation metrics exist (silhouette coefficient, Calinski-Harabasz index, Davies-Bouldin index, Dunn index, etc.); these metrics are referred to as clustering validation scores. All the metrics are based on differences or ratios of distances within clusters versus distances between clusters (e.g., Xu and Tian 2015).

External evaluation compares competing clustering methods to a similarly labeled dataset where the true grouping is known and then sees how well each method finds that ground truth; this relies on information (namely, the labels assigning each instance to its correct cluster) that is not part of the dataset. A comparison to an external benchmark is ideal, but it relies on information that is often unavailable. Having a metric for clustering quality does not entirely solve the problem of assessing algorithm or model quality. It may not give insight into the general performance of the clustering algorithm. Additionally, it may not capture the operational usefulness. Lacking ground truth, it can be hard to distinguish between poor algorithm performance and data that cannot be cleanly separated by any algorithm. If ground truth can be determined for a small sample of data points, it is possible to compute a confusion matrix (as with SL) for that subset. It can also be useful to test algorithms against a benchmark to see how they perform on similar problems, in both absolute and relative terms.

For the particular problem of clustering, algorithms should have a few specific robustness properties:

- The addition of a small amount of random noise to the input should not cause major changes in either the number or composition of clusters.
- The presence of a few erroneous data points or other extreme outliers should not affect overall algorithm performance.
- Transformations of the data (such as converting categorical data to numerical scales) should not cause major changes in either the number or composition of clusters.

For each of these dimensions of robustness, the model’s performance on the given dataset can be assessed (at least qualitatively—the meanings of “major” and “few” vary from mission to mission) by perturbing the data and comparing the output of the model to the original output by adding random noise; adding incorrectly formatted/incorrect data; expressing attributes in different units on different scales, etc.; or for complex data modalities, applying a different embedding. The amount of change that is acceptable will depend on the application.

3.2.4.6 Reinforcement Learning Metrics

RL is a means to learn how to achieve a well-defined goal by discovering a policy. This is done through an agent (an entity that will perform an action) that will interact with an environment and is rewarded for making “correct” decisions (and punished for incorrect ones). The policy specifies, for each state the system can be in, what to do if the system is in that state. Although there are several types of RL algorithms (model-, policy-, and value-based), the end goal for any RL algorithm is to maximize its total reward while avoiding penalties. The reward function itself is thus the natural metric for model performance—if a better metric were available, it would also constitute a better reward function.

The exact form of an RL reward function is highly dependent on the specific application, but several distinct approaches exist for designing reward functions. The cumulative reward (either over the entire lifetime or per episode) is the most straightforward metric. The average reward per time can also be calculated. Another measure of reward is the discount reward that adds a discount factor to future rewards. The choice of discount factor can alter the priority of immediate rewards versus long-term rewards; this decision will be influenced by the mission and operational environments (Pullum 2022).

3.2.5 VV&A of Machine Learning Models

In accordance with DoDI 5000.61, models used in defense processes will be subject to VV&A. As with data VV&A (see Section 3.2.2.3), the military Components will define procedures and standards for VV&A of models. At the time of this writing, no such published procedures or standards have been defined for VV&A of ML models. Section 4.6.6 provides additional discussion of possible DT&E support to ML model VV&A activities.

3.2.6 Model and Data Drift

This guidebook provides several metrics to evaluate a model's performance. It is important to note, however, that model performance is typically not static. In particular, model performance tends to change over time as the operational environment changes (or is changed). This could be due to the use of the model in a new context or for a new mission; to adversary responses to blue force tactics; or to general changes in the world. Degraded performance due to such changes is called "model drift," "concept drift," or simply "drift" (Webb et al. 2018). It is important to continually monitor model performance using updated TVT data, and changes in performance should be investigated. By assessing changes in data distribution over time, the source(s) of drift can be identified.

Drift is typically detected and measured using feature-based methods that compare the distribution of recent operational inputs to that of the TVT data (e.g., Webb et al. 2018). The ability to detect and measure drift may depend on instrumenting the ML model inputs and outputs in ways that are not operationally necessary for any one mission. DT&E can inform system designers of the potential value of such instrumentation, to be compared against the design margin claim of such instrumentation.

Drift can prompt the need to retrain the model, collect more data, or generate new synthetic data to capture changes in mission or environment. DT&E activities, even post-fielding, can be useful in planning for the possibility of drift, monitoring for drift, detecting drift, and quantifying the impacts of drift.

3.3 Modeling and Simulation for DT&E of Machine Learning

Needs for and Benefits of Using Modeling and Simulation

Regarding AIES/MLES, the Defense Science Board Summer Study on Autonomy states the following:

> Insuring that the system will respond appropriately to all of the possible inputs will exceed the capability of conventional testing. It will require using a combination of modelling and simulation to explore thousands of test cases, statistically measuring system performance against the desired standard, then doing real world testing of the system to ensure that the modelled and real world behavior match for corner cases that span the range of system performance.

Others have similarly concluded that live testing alone will be insufficient to conduct T&E of AIES/MLES given their potential for continual change due to learning and/or potential brittleness and lack of robustness (Koopman and Wagner 2018).

M&S can, at least in principle, be used to conduct DT&E of AIES/MLES under conditions and scenarios not possible for live testing. Such conditions could include very hazardous or life-threatening situations and complex, proliferated threats. The use of M&S to generate and test numerous scenarios digitally that could not—because of their sheer numbers and/or complexity—be explored through live testing can also enable the discovery of “corner cases” in which the AIES/MLES exhibit undesired and/or emergent behavior that would otherwise go undetected. M&S can also be used to generate synthetic TVT data to complement and/or augment the data otherwise available for developing ML models. Care must be taken, however, to perform the evaluation of those data needed to ensure they contain the key features of the operational domain the ML models are using. That evaluation will likely include live testing as well as SME and review. The M&S-generated data could also incorporate modifications that could cause pernicious learning, thereby enabling evaluation of potential adversarial attacks.

Challenges

The use of M&S for T&E of AIES/MLES, however, will involve numerous challenges. The highest fidelity M&S of these systems will use the system’s actual software operating in real time, fed by realistic inputs provided by digital models of the system’s sensors (or hardware operating in the loop), as well as simulations of the environments in which those sensors will operate. Any approach other than the actual software running on the intended operating system and processors risks missing important failure modes. Such M&S will require the government to have purchased the rights to the software and to maintain, either itself or through contracts, the expertise needed to use the software as it is modified and to integrate it within the M&S.

M&S used to conduct T&E of the ML model as a stand-alone may be less complex than the simulations used for the integrated system. That M&S, however, still needs to mimic the inputs and outputs to and from the ML model when used in the integrated system with some degree of realism; otherwise, the M&S results will have limited utility for evaluating the performance of the ML model.

All of the M&S used for creating synthetic data; simulating the stand-alone ML; and simulating the integrated system's operation including its sensors, environmental inputs, and actuator responses will require VV&A. VV&A will likely be a challenge for several reasons:

- Unknown features of the real environment missing in the M&S environment (or M&S-generated training data) could cause failures or otherwise drive inaccurate simulated system behavior. This is particularly dangerous for ML models given potential sensitivity to high-dimensional features of data that may be irrelevant to humans or traditional algorithms.
- Inaccuracies in the timing of sensor and ML inputs and outputs in the M&S could result in inaccurate simulated behavior. ML systems are often sensitive to high-dimensional features irrelevant to humans or traditional algorithms. This can introduce new sources of error or inconsistencies.
- The data available from live testing are likely to be a small subset of the inputs and environmental conditions of interest for use with the M&S. This may exclude aspects of the actual operating environment of interest (e.g., complex, proliferated threats).[11]
- Important features of the real environment, such as the stochasticity inherent in the ML algorithms, sensor environmental inputs, actuator responses, and computational timing, may be hard to incorporate in the M&S. AI/ML systems pose additional difficulties because an unknown quality of the dependencies is introduced into otherwise well-modeled behaviors.

Building this M&S and performing its VV&A could therefore be a lengthy, complex, and expensive technical effort in and of itself. The M&S used for the fully integrated system is also

[11] The M&S Guidebook discusses methods that can be used for VV&A when live test data are a small subset of M&S data. These methods, however, cannot provide VV&A across regions of the operating space for which live test data are absent and system behavior may be highly nonlinear.

likely to be computationally challenging. M&S "inevitably involves a tradeoff of fidelity vs. runtime cost as well as questions about completeness and accuracy of software models. Simulation suffers from the possibility of not simulating unanticipated scenarios (e.g., *unknown* safety-relevant rare events)" (Koopman and Wagner 2018, 3). Constructing a series of M&S operating at increasing levels of fidelity used with live testing may be one way forward.[12] Higher fidelity M&S could be used to "validate the correctness of lower fidelity models, but must also be explicitly designed to emphasize checks of the assumptions and simplifications that are known to be present as simulations are run" (Koopman and Wagner 2018, 6).

Other approaches to using M&S to define and limit the scope of T&E are being attempted. One such approach decomposes the missions that an AIES/MLES is meant to perform into tasks and subtasks and then evaluates the resulting behaviors that the system exhibits using Type 2 fuzzy logic (Dalpe et al. 2021). Another approach attempts to determine the regions of the operating space most relevant and critical to testing and evaluating performance of the AIES/MLES. "Instead of modeling the underlying behavior of a black-box autonomous system, we try to model the relationship between scenario configuration and the resulting performance metrics for the autonomy" (Mullins et al. 2017). However, these approaches also appear to be relatively complex to implement.

Given these challenges, planning for the application of M&S to the T&E of AIES/MLES, including approaches, schedules, and resources, should begin as early as possible in the system's development and be updated continually. Development and use of the M&S will likely proceed iteratively in tandem with the development of the system itself. Close collaboration between testers and the AIES/MLES developers will be required for success.

3.4 Use of Formal Methods in AI

Formal methods require skill sets independent of AI technology. If developers have used formal methods to develop a system with an AI component, the T&E community will need familiarity with those methods to evaluate whether they were properly applied. If the T&E community employs formal methods to test and evaluate a system with an AI component, its members will need more than familiarity with those methods: They will need expertise. Furthermore, many formal methods are best applied using automated tools. The T&E community must be able to obtain and install those tools.

[12] The architecture of the AIES/MLES may also affect the fidelity of the M&S needed. For example, an actuator-monitor/governor architecture may permit high-fidelity M&S of the monitor and lower fidelity M&S of the ML-enabled actuator.

3.4.1 Overview of Formal Methods

Formal methods employ rigorous approaches to analyzing systems. They use mathematics and logic to determine whether an artifact satisfies some set of well-defined properties. Formal methods rigorously evaluate how software will behave in a wide range of scenarios, including edge cases, that cannot be observed through unit testing or simulations. Formal methods applied to design specifications can allow developers to anticipate and avoid errors early in the software development process.[13]

There is no universally accepted definition of the term "formal methods." For the purposes of this section, a formal method necessarily has a mathematically rigorous basis. This basis may use set theory, logic, proof techniques, discrete mathematics, probability and statistics, topology, and other quantitative fields of study. A formal method is repeatable: The application of a formal method can be examined to verify whether the method has been correctly applied and its conclusion is valid. Where formal methods apply, they establish a higher level of confidence in their conclusions than is obtainable through empirical testing or evaluation of the development process.

As depicted in Figure 3-3, formal methods require a statement of desired properties, some entity to be assessed against those properties, and a basis for using the formal methods; the result is a statement of whether the entity conforms to the properties. The formal method can be performed manually or automatically.

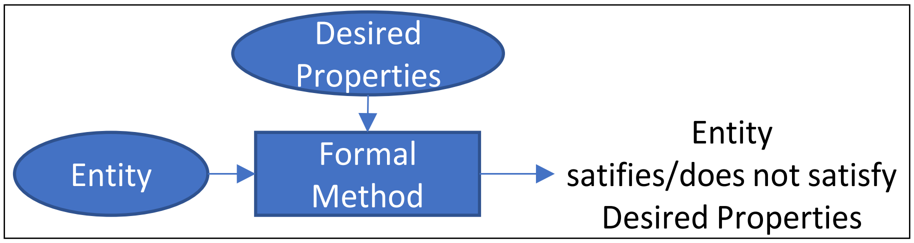


Figure 3-3. Formal Methods Concept

Examples of modern formal methods include:

- Static Analysis. Static analysis attempts to detect errors in programs such as undefined variables and invalid type conversions that usually stem from careless programming and typographical mistakes and are typically revealed the first time the statement in which they occur is executed. Static analysis can also attempt to detect more subtle errors such as unreachable code, which may not be revealed through execution except through a test

[13] For an overview of formal methods, see the *Concise Guide to Formal Methods: Theory, Fundamentals and Industry Applications* (O'Regan 2017). For an overview of formal methods in machine learning, see "A Review of Formal Methods applied to Machine Learning" (Urban and Miné 2021).

case that attempts to exercise the unreachable functionality. Most modern compilers perform static analysis. The types of analysis they perform depend on the programming language; strongly typed languages such as Java and C++ permit more analysis than dynamically typed languages such as Python. There are also specialized static analysis tools that run independently from compilers; an Institute for Defense Analyses study of DoD projects, "Cybersecurity and DoD System Development: A Survey of DoD Adoption of Best DevSecOps Practice," found that many use Fortify,[14] SonarQube,[15] or both (Kuzio de Naray et al. 2021). These typically perform computationally intensive static analysis that would be too time-consuming to execute during every compilation operation.

- Model Checking. Rather than applying formal methods to a production system, developers create a model of some aspect of the system and then apply formal methods to that model. For example, the SPIN tool (Holzmann 2003) is used to verify the correctness of concurrent software systems. Sometimes a model is used to generate working software (Raistrick et al. 2004).

- Deductive Verification. Deductive verification relies on an axiomatization of a programming language and the use of deductive logic based on those axioms to reach conclusions about a program (Hoare 1969).

- Proof Assistants. Proof assistants are automated tools to assist in formally proving properties. Manually performing proofs is labor-intensive and error-prone (Greiffenhagen 2024). Tools that can perform some or all of the steps help make proofs of larger systems practical. Proof assistants have been used to perform deductive verification (Geuvers 2009).

- Model-Based Testing. Model-based testing uses tools that automatically generate test cases for a system based on a formal model of the input space, the output space, and a mapping between them. Model-based testing tools require system execution, but they formalize expected system properties (input and output) and are therefore considered to fall within the realm of formal methods (Schieferdecker and Hoffmann 2011).

Formal methods are often thought of as applying to executable software, but formal methods designed to analyze ML models and datasets have also been developed (Urban and Miné 2021). Dataset-specific formal methods typically check syntax rather than semantics. For example, the XML Document Type Definition and XML Schema specifications were early approaches to verify whether an XML document was sent and received in a format conformant to an expectation. Some data-oriented formal methods support deductive semantics. The Resource

[14] https://www.microfocus.com/documentation/fortify-static-code/.
[15] https://www.sonarsource.com/products/sonarqube/.

Description Framework Schema (Gandon et al. 2011) supports a form of deduction in which data expressed as a graph *entails* another graph with additional edges; these new edges add knowledge. The Web Ontology Language (OWL) (see the W3C OWL Web Ontology Language Overview Website at https://www.w3.org/TR/owl-features/) supports description logics.

3.4.2 Formal Methods and the AI System Development Life Cycle

ML systems and non-ML systems[16] are developed differently. Software for a non-ML system is developed using a specify-design-code-test iterative life cycle. Formal methods can be applied throughout an iteration. By contrast, software for an ML system necessarily incorporates data, and the system is not operational, or even testable, until it has been sufficiently trained on some dataset; see the notional AI system development life cycle in Figure 3-4. The traditional concept of applying formal methods—that, so long as a specification of desired semantic properties exists, formal methods can be applied throughout the life cycle—is meaningless in a life cycle where the semantics do not emerge until after extensive system execution and where a system's behavior may have more to do with the training dataset than with the algorithms that use that dataset as input. Section 2.4 discusses the implications of planning for T&E of AI systems. These implications extend to formal methods, where repeatability is doubtful: Scraped data may cease to be available, in which case the prepared data cannot be recreated. Even if the prepared data were recreated, the formal methods available for analyzing the prepared data (see Sections 3.4.5 and 3.4.6) lack the predictive power of formal methods applied to executable code.

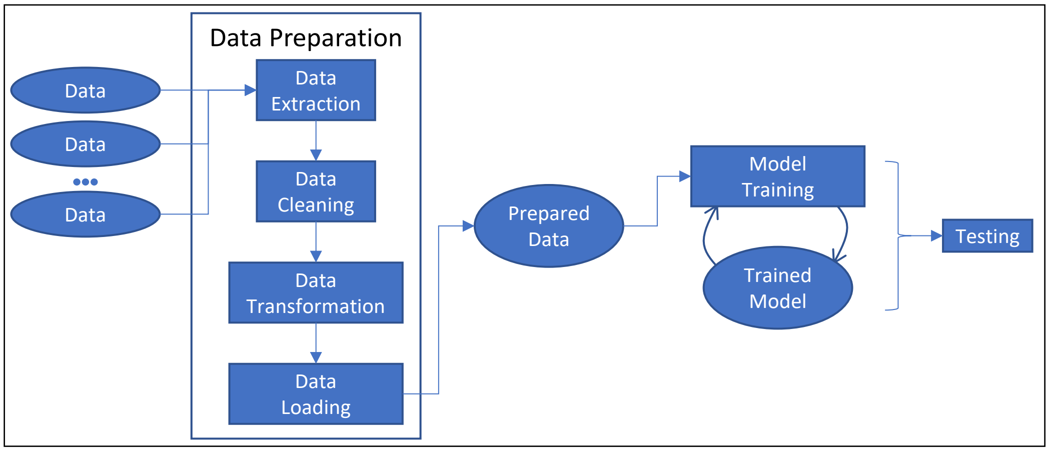


Figure 3-4. Notional AI System Development Life Cycle

[16] In this context, a non-ML system means a system that does not require training before deployment. This would include some rule-based systems, which some consider to be AI systems.

3.4.3 Formal Methods Applied to AI System Components

Rectangles in Figure 3-4 represent executable systems. Each system has properties independent of the “AI” purpose of the ultimate system. The five formal methods—static analysis, model checking, deductive verification, proof assistants, and model-based testing—can be separately applied to each system. The properties of the data transformation system, for example, are describable: Given its expected inputs (which the data cleaning system prescribes), does it produce outputs in the format expected by the data loading system? Even the model training system can be subjected to formal analysis. Suppose it implements a neural network. What are the expected properties of its activation function? Are layers properly connected? If neurons are implemented as concurrently operating subsystems, model checking tools like SPIN (Holzmann 2003) might be useful to verify that the system never deadlocks.

Activation function behavior, network connectivity, and deadlock avoidance are properties of formal methods that can be used to establish executable code. Those properties do not apply to ML models, whose executable logic concerns abstract concepts such as matrix multiplication. It may be possible to prove things about a neural network’s behavior based on its possible outputs, that is, the neurons in its output layer. In a safety-critical system, the combination of some set of neuron activations in the output layer might be known to cause a hazardous state. If analysts can identify such hazardous states, formal methods can be used to demonstrate that the hazardous state will *not* occur (Bunel et al. 2018). These methods can be split into several categories:

- Satisfiability Modulo Theory, which abstracts the problem into a constraint satisfiability problem.
- Mixed-Integer Linear Programming, in which safety is considered as a mixed-integer linear program.
- Global Optimization, which is useful for analyzing the instability of neural networks, that is, their susceptibility to small perturbations (Szegedy et al. 2014).

Methods in each category are best suited to certain types of neural networks, so no approach can be said to be superior to others. Tools implementing these methods have been executed on benchmark networks and have demonstrated their feasibility for certain real-world use. The Neurify tool was able to verify safety properties of a 10,000-neuron network (Wang et al. 2018). This size network, although useful in some safety-critical applications, is still orders of magnitude smaller than the sort that would be used for an LLM such as would be derived using a dataset such as the Pile (Gao et al. 2020).

3.4.4 Formal Methods and Real-Number Arithmetic

Most formal methods are based on real-number arithmetic, not the approximations computers use. Many ML models use relatively low-precision floating point representations. Any formal method analysis of an ML model necessarily involves approximations that, propagated over time, can introduce errors (Li et al. 2019). This is a recognized problem for all computational software; the field of numerical analysis is dedicated to quantifying and minimizing such errors. Some ML-specific work has been devoted to dealing with the problem. For example, in “An Abstraction-Refinement Approach to Verification of Artificial Neural Networks,” Luca Pulina and Armando Tacchella (2010) describe how to use formal methods to determine whether the output of a neural network is always within specified bounds. Their approach is limited to a feed-forward fully connected neural network with sigmoid activation functions, restricting its practical application. Subsequent work (Pulina and Tacchella 2012; Scheibler et al. 2015) has encountered scalability problems.

3.4.5 Formal Methods in Data Preparation

The previous sections focus on formal methods in the later stages of the AI software development life cycle shown in Figure 3-4. Researchers have also considered how to apply formal methods during data preparation activities.

Data preparation activities sometimes, although not always, employ AI-based tools. Data extraction activities may involve scanning document images and performing optical character recognition (OCR) or recognizing and interpreting charts, graphs, pictures, and tables. OCR, an early AI success story, albeit one whose foundations long predate computers—its statistical basis can be traced as early as 1914 (Dhavale 2017)—transforms an image into text in some character encoding system such as ASCII or Unicode and is by now a well-understood topic. Developers have available both commercial and open-source products to perform the task, and accuracy is high. Tools to recognize charts, graphs, and tables are not as well-developed, but they exist. The difficulty in interpreting a chart or graph is that it requires not just interpreting the data but also understanding the axes. Graphs sometimes have axes labeled “x” and “y,” the meanings of which appear somewhere in surrounding paragraphs. Because of these kinds of complications, there is no general approach to understanding how to interpret a graph.

Data extraction that does not involve image scanning does not have to use AI. Some data extraction simply entails querying a database. Data extraction can also involve parsing semi-structured data such as a spreadsheet; syntactic analysis can be used to the degree the data source permits. Sometimes data extraction means web page scraping, which involves parsing Hypertext Markup Language (HTML). HTML-embedded text can be extracted using open-source tools and libraries. Tables in an HTML document are easily identified by HTML tags, as are their rows and columns. In these cases, developing an algorithmic extraction tool that understands the syntactic rules of HTML tables is probably quicker and cheaper than training an ML system. (Extracting information from images embedded in HTML may require AI.) Traditional formal methods can be used to verify the correctness of these extraction tools. In this scenario, formal methods are being used in the AI software development life cycle, if not necessarily on AI systems.

Organizations implementing an AI pipeline (the term for the sequence of data preparation and model training activities in Figure 3-4) should consider which activities would benefit from the application of AI and which can be implemented algorithmically. For example, data cleaning of text can be as simple as checking for spelling errors or as complex as using an LLM to assess whether the grammar of a sentence is correct. The decision to use AI in an activity may require an embedded life cycle to train that system; the cost-benefit analysis for such a decision is outside the scope of this guidebook. It is, however, relevant to mention that embedding a life cycle may benefit from using the sorts of formal methods discussed here. Their availability, and the value they add, should be part of the cost-benefit consideration.

3.4.6 Formal Methods for Detecting Dataset Bias

The success of an AI application depends in large part upon the dataset on which the application is trained and whether that dataset exhibits unwanted bias. Bias can cause spectacular AI system failures.

Researchers recognized bias in datasets long before the rise of data science and ML-based AI, and they have been working on methods to detect and mitigate bias ever since. They have proposed both qualitative and quantitative methods (Nagubandi 2021). The qualitative methods provide more of a framework than a rigorous approach, often serving as checklists or processes to follow to ensure that a dataset's characteristics have been properly considered—ensuring that the stated purpose of the dataset aligns with the objectives of model training; that the collection methods used are sufficiently rigorous; and that the integrity of tools used in data collection and storage is adequate, among other considerations.

Although these approaches are important, they are not formal methods. The quantitative methods, which have a basis in statistics, can be considered formal, even if they yield probabilistic answers rather than the black-and-white ones that most formal methods produce. These methods are designed to both measure and mitigate bias. Many exist as automated tools that offer data scientists options to analyze datasets, visualize whether a dataset is skewed toward certain populations, and mitigate bias. Example tools include:

- Fairlearn, an open-source Python-based programming suite whose objective is "to help data scientists improve fairness of AI systems."[17]
- IBM's AI Fairness 360, another open-source toolkit, with application programming interfaces (APIs) in both Python and R.[18]
- Google's Fairness Indicators toolkit, targeted toward datasets used for ML.[19]

As with any statistics-based approach, each tool has strengths and weaknesses. This can be summarized by noting that every statistical model makes assumptions about data, and using a model on data that violates those assumptions can lead to unpredictable results. Analysts with strong data science skills must determine which tools to apply to a given dataset and how to apply them. This requires using the qualitative methods outlined above.

3.5 Visibility into Machine Learning Models

The inner workings of many modern ML techniques are not interpretable by humans, meaning that a human cannot know why an ML model gives the answers it does. Additionally, ML models tend to be brittle, meaning that they can be sensitive to changes in input that are undetectable by humans, and they can produce very different outputs based on small changes in input. This sensitivity to small changes in input, together with large input and output spaces, means that traditional methods of T&E may not be sufficient to characterize system capabilities, limitations, and risks. This section discusses new methods that are used to build assurance about system behavior. These methods include diagnostic tools and XAI techniques, which have been developed to help humans interpret and understand the behavior of various models. The results of such methods are collectively known as visibility into ML models.

3.5.1 What Is Visibility?

Visibility into an ML model refers to access to information about the model's decision-making processes. This access ideally enables the relevant stakeholders, beyond the development team, to understand what causes a model to produce the outputs it does—and in particular, under what circumstances the model is trustworthy for its intended purposes. The level of visibility should be sufficient to enable stakeholders to assess model performance, reliability, and generalizability to the extent necessary.

There is currently limited theoretical understanding of how ML models work, and that understanding has yet to generate practical tools for characterizing ML model performance

[17] https://fairlearn.org/.
[18] https://github.com/Trusted-AI/AIF360.
[19] https://www.tensorflow.org/.

envelopes. In general, users and stakeholders will not know exactly how any given model can be expected to behave. In particular, the parameters of nonsymbolic ML models do not have natural interpretations, which means it can be challenging to predict whether a model will be dependable when applied to specific inputs. Lack of theoretical understanding of the model or natural interpretations of the parameters, together with sensitivity to small changes in inputs, forces a new approach to testing.

Despite the difficulty of such testing, it is critical that the relevant authorities understand both risks and benefits of when a model will and will not be dependable. This creates a need for methods to establish visibility into and determine the generalizability of the model. If a model has good visibility, then developers and testers can use visibility early in development to understand the cases in which the model is likely to fail; this allows them to improve the model performance on the TVT data and improve its performance on new data, potentially to include improved ability to fail safely. Later in development, visibility helps determine the generalizability of the model and predict the cases in which the model will be useful.

Several tactics have been developed to provide humans with visibility into the behavior of various models. These tactics, described in Sections 3.5.3 and 3.5.4, are intended to help understand which input features drive model behavior and to determine the situations in which a model will be dependable enough for the intended use case. Some of these tactics are specific to particular model types, whereas others can be applied as model-agnostic methods. Collectively, these approaches provide visibility into ML methods. Visibility tools can be useful both for calibration of user trust and for T&E, including diagnosis of system behavior and characterization of system limitations.

3.5.2 Why Visibility Matters

Any AI model must be sufficiently trustworthy for its intended use. The three components to a model’s trustworthiness are as follows:

- When employed correctly, the model will dependably do well what it is designed to do.
- When employed correctly, the model will dependably not do undesirable things.
- When paired with the humans it is intended to work with, the model will dependably be employed correctly.

An assurance package is a collection of arguments that a system is sufficiently trustworthy for its intended use case. The contents of an assurance package will depend on whose trust is needed; the level of confidence required for the potential risks and benefits; and the level of confidence justified by the available evidence.

Visibility helps produce evidence that contributes to assurance. Visibility will guide testing to provide the maximum mitigation of the challenges to trustworthiness such as sensitivity to inputs and the lack of underlying theory.

Visibility is helpful in other areas as well:

- Generalizability. Visibility helps developers understand how a system is likely to generalize to different datasets or use cases. In a model that is expected to be deployed widely on different datasets, visibility is key to being able to provide evidence that the model will generalize beyond the first dataset for which it was developed.
- Model drift. As discussed in Section 3.2.6, model drift is a common problem in deployed models. Mitigating model drift is easier in models with a high level of visibility because visibility allows developers to better understand the causes of drift and therefore how to lessen it.
- Risk management. The biggest risk factor with many ML models is a lack of understanding of the situations in which they will be reliable. Because many models are quite brittle, understanding the limits of generalizability of a given model is essential for managing risk, and visibility is key for understanding those limits.

3.5.3 Diagnostics

The number of test cases for a model is often small relative to the size of the possible input space. This means that it is possible for a model to work well on the test cases by chance. For stakeholders to gain confidence that a model is reliable, rather than simply “getting lucky” on a handful of test cases, they need to be assured that the model is making decisions for the right reasons, not just making decisions with acceptable outcomes by coincidence.

Diagnostic instrumentation and tools are crucial for providing visibility into a model’s decision-making processes. Diagnostics and telemetry are often used in development but removed for fielding. In AI applications, however, retaining runtime diagnostic instrumentation of the model throughout the life cycle of the system may be desirable, despite the added cost in weight, power, or cooling requirements. ML models are usually expected to change more frequently over the system’s life cycle than traditional system components, often in unpredictable ways. In particular, every time a model is retrained with new data, its performance will change. Sometimes the resulting change in performance will be significant, and sometimes it will be minor, but it is not generally possible to predict the type or magnitude of the change in performance that will come from retraining. Given this uncertainty, it is essential to retain the diagnostics to correct the model after deployment if its performance deteriorates.

The following subsections describe four different diagnostic approaches to XAI:

- Input-based XAI strategies.
- Local interpretable model-agnostic explanations (LIME).
- Existing software test methods that can be adapted to testing ML.
- Runtime monitoring of AI.

Each of these approaches can generate diagnostic evidence that contributes to an assurance package evaluating the extent to which an ML model is sufficiently trustworthy for its intended purpose.

3.5.3.1 Input-Based XAI Strategies

One approach to interpreting model behavior is to examine which parts of a given input have the most influence on determining the output. This approach is called *input-based XAI*. In some cases, this information provide insights into how the model is making its decisions. Understanding the most influential parameters of a model offers clues to the generalizability of the model. Researchers have developed various metrics to identify the input values that are most influential in determining the outputs of an ML model:

- Drop column: If the model is retrained leaving out one parameter at a time, how much does the output change? This change can be computed for a given input or average over the entire test set.
- Saliency map: If the partial derivative of the output is estimated with respect to each input parameter, which parameters have the steepest gradient? These are the “salient inputs.”
- Shapley additive values: If all possible models that do not use the kth parameter are fitted, on average how much does the prediction change when the kth parameter is added to the model and the model is retrained? (This is equivalent to doing “drop column” on all submodels as well as the full model.)
- Permutation: For each input parameter in turn, if a model is fitted to training data where that parameter’s values are replaced by randomly assigned values from instances in the test set, for which parameters does this have the most effect on predictions?

3.5.3.2 Local Interpretable Model-Agnostic Explanations

Another well-researched method for interpreting neural network model outputs is called *Local Interpretable Model-Agnostic Explanations,* or LIME (Ribeiro et al. 2016). LIME works by fitting simple linear regression models for inputs near the input to be explained. (That is the "local" part of the name.) The regressions are performed on perturbed input data, producing a sample of points "similar to" the input in question along with the model's outputs for those points. For images or text, perturbations might include leaving out certain pixels or words or replacing them with random values. The slopes of these local regression models have a similar interpretation to the gradients estimated in the saliency maps described in the previous section.

Because it uses regression, LIME works best on models that produce a continuous numeric output rather than a binary or categorical output. LIME is broadly applicable because it is model agnostic; it can be applied to any model with quantitative output, whether symbolic or subsymbolic, because it requires only perturbed inputs and their corresponding outputs.

3.5.3.3 Software Test Methods Adapted to Machine Learning

Most software testing relies on *test oracles*, which are specifications of what the correct output is supposed to look like for a given input. For MLES, it is often difficult to determine whether a given output is correct or not, even for inputs similar to those in the TVT set. Also, as noted earlier, smooth changes in output between test points are not necessarily expected. Despite these challenges, some software test methods can be applied effectively to MLES.

Metamorphic testing is a software test method that can be used in some cases when no test oracle is available. The key insight is that often, even if the correct output corresponding to an input x is unknown, it *is* known how the output should change if the input x is perturbed in a specific way (Murphy et al. 2011). The two versions of this approach are as follows:

- Known change upon perturbation: The type of output change to expect is known if the input is changed in a particular way.
- Stationary upon perturbation: Changing the input in a particular way should not make any difference to the output.

3.5.3.4 Runtime Monitoring of Machine Learning

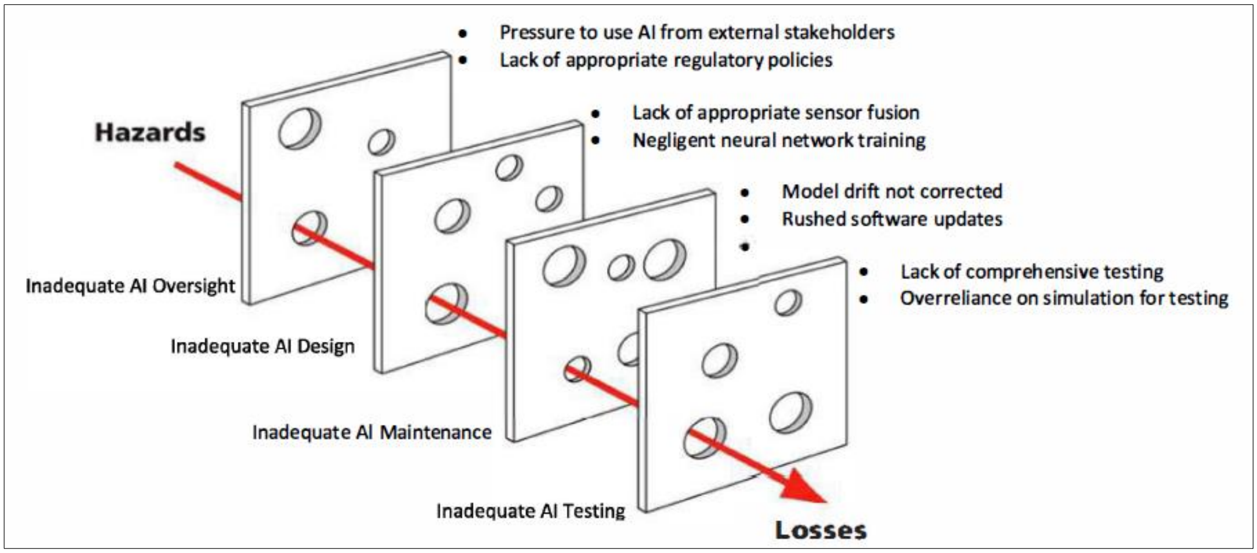
Diagnostic instrumentation is common in many engineering domains. Modern automobiles, for example, continuously monitor many aspects of engine status and performance. Technicians can access these data through defined data ports that are permanently installed on the vehicle. Commercial aircraft continuously transcribe system data to a flight data recorder and cockpit voice recorder for use in diagnosing faults in the event of a mishap. In both cases, the value of the monitoring systems has led designers to allocate to them the necessary space, power, and weight in the system design budget.

Similarly, it is possible to monitor the inputs, outputs, and (in some cases) internal behavior of ML models. These measurements might be stored internally (as with a flight data recorder) for later retrieval, transmitted as telemetry information, or accessible only by testers with the appropriate support equipment (as with automotive data), depending on the application and assurance needs of the AIES.

Some international standards implicitly call for runtime monitoring in support of safety-critical functions in intelligent systems. For example, UL 4600, "Standard for Safety for the Evaluation of Autonomous Products," calls for "field engineering feedback to manage a continuous improvement process to identify and mitigate risks due to a changing, open world environment as well as encounters with unforeseen heavy-tail events." For such feedback to be effective in diagnosing failure mechanisms, it would need to include significant information about the system state at the time of an incident, which could be provided only by runtime monitoring. If developers implement runtime monitoring for an AIES, access to the data collected would have obvious utility for DT&E, as well as for the certification activities discussed in Section 4.6. Testers can also advise systems engineers regarding the potential benefits (in cost, schedule, and risk mitigation) of instrumenting the AI for specific information.

3.5.3.5 Prevention and Diagnosis of Failures

The expanded role that T&E can play in preventing failures when AI makes or participates in decision making during system operation, as well as in comprehensively diagnosing the failures that will inevitably occur, is illustrated by failures that have occurred during the operation of self-driving cars (Cummings 2024). One failure resulted in life-threatening injuries to a pedestrian dragged by a self-driving car executing an unnecessary and inappropriate maneuver. Another failure resulted in a collision between a self-driving car and an articulated bus, a type common in San Francisco where the car was operating. Analyses of these failures have been conducted by applying a Taxonomy for AI Hazard Analysis. The taxonomy adapts a Swiss cheese model of human error (Reason 2000) to systems in which AI performs or participates in decision making. The premise is that the multiple layers of defenses employed to prevent failures actually each have holes in them like Swiss cheese through which hazards can escape. These holes are continually moving, appearing, and disappearing but occasionally align and cause a failure; see Figure 3-5.


Pressure to use AI from external stakeholders
Lack of appropriate regulatory policies
Lack of appropriate sensor fusion
Negligent neural network training
Model drift not corrected
Rushed software updates
Lack of comprehensive testing
Overreliance on simulation for testing
Hazards
Inadequate AI Oversight
Inadequate AI Design
Inadequate AI Maintenance
Inadequate AI Testing
Losses

Source: (Cummings 2024)

Figure 3-5. Taxonomy for AI Hazard Analysis Applied to Failures of Self-Driving Cars

In these two cases, the analyses find that failures occurred at each level in the AI taxonomy in the computer vision neural networks employed by the cars to understand their operational environment:

- Inadequate AI Oversight: no state-mandated standards for autonomous vehicle design and operations.
- Inadequate AI Design: reliance on neural networks incapable of forecasting derivative outcomes such as the consequences of a pedestrian stepping in front of another vehicle with which the self-driving car was entering an intersection.
- Inadequate AI Maintenance: lack of updates to roadway maps indicating the correct number of roadway lanes.
- Inadequate AI Testing: lack of testing exploring comprehensive operations across all safety-critical features of the operational domain.

The company involved in both failures (Cruise^{tm}) essentially shut down its operations following the pedestrian injury, illustrating that the consequences of failures in systems employing AI can be substantial.

As illustrated by the previous sections of this guidebook, advice from the T&E community on the processes used to develop and implement AI models, as well as the results of the T&E conducted on the datasets, stand-alone AI models, and AI models embedded in systems, can identify problems across all slices of the taxonomy:

- Inadequate AI Oversight: for example, lack of policy-mandated AI design features or restrictions in the concept of operations whose absence resulted in failures.
- Inadequate AI Design: for example, training not covering all aspects of the operational domain.
- Inadequate AI Maintenance: for example, unaddressed concept drift causing operational failures.
- Inadequate AI Testing: for example, key aspects of the operational domain not covered in live testing or realistically represented in M&S.

The T&E community can make significant contributions not only to diagnosing failures but also to avoiding them throughout the design, development, and operation of systems employing AI.

3.5.4 Traceability

Traceability refers to the ability to reconstruct how an ML algorithm came to behave the way it does. The requirement for traceability as stated in the DoD AI Ethical Principles is that "AI capabilities will be developed and deployed such that relevant personnel possess an appropriate understanding of the technology, development processes, and operational methods."

Any well-governed AI process will come with accompanying model and data cards, which can be used to support traceability. The model card should describe what the model is; what it is supposed to do; and what shortcomings or problems it is known to have. A well-written model card should be sufficient to trace model performance from system requirements to output.

Similarly, the data card should provide an overview of the training data and the data flows for the model. T&E teams should be able to use the data card to understand the baseline data and trace features of the training data to the model's outputs.

Traceability does not come automatically with most ML models, and additional efforts are required to establish it. The testing strategy for establishing traceability should include plans and documentation covering development methodologies, training data provenance, ML model instrumentation, and planned uses of XAI techniques.

3.6 Early Involvement

The need to "shift left" in both DT and OT has been discussed for years. The most obvious ML-related need for early involvement—assessment of the availability and quality of TVT data—is addressed in Section 3.2. Early assessments of technology risk and technical feasibility are covered in Sections 2.2.2.3 and 2.2.3.1. These are essential elements in an ML development life cycle. In iterative life-cycle models, such as Agile and DevSecOps, these assessments will take place early in each iteration.

A primary motivation for early involvement is to reduce the discovery of problems during DT, OT, or integrated activities late in development. This section is specifically focused on how to inject an operational or mission-oriented focus into very early model testing and design exploration that can reduce the probability of late discovery of problems (and mitigate the impact of them).

The CDAO OT&E of AIECs framework specifically acknowledges that "operational testing is constrained by limited resources, such as time, money, expertise, and infrastructure. As a result, it is not feasible for OT&E to completely cover the operational envelope." This applies not only to formal OT but also to any testing of complete systems in realistic scenarios. The framework also points out that "it can be difficult, if not impossible, to evaluate which operational conditions are likely to cause system failures or undesired behavior."

Although a broad set of exploratory operationally realistic tests may be unaffordable, and determining which operational conditions will lead to system failures is challenging, the problem can be approached from the perspective of key elements of the technologies used to deliver the desired capability. It is feasible to identify key specific components or elements of the design that are most important to delivering capability and most likely to be stressed by the operational conditions.

For hardware systems, this is akin to the TRA approach to controlling certain types of risk by asking the following questions: What technology performance is needed to satisfy requirements? What if any of this performance is new and has not been demonstrated?

In the context of AI systems, the focus should be on identifying which elements are likely to be stressed in operation. Expertise informed by the understanding of the AI technology and the mission context makes it possible to identify technical drivers of the performance that are most sensitive to these challenges. The specifics will be system dependent. Areas specific to AI that are likely to be vulnerable to system-specific stresses warrant expanded early characterization testing. The following subsections identify and discuss key features of three of these areas:

- CONEMP.
- Characterization of average versus outlier behavior.
- Reward hacking.

3.6.1 Concept of Employment

The interplay between system design and CONEMP may warrant early extensive characterization. Early design efforts can provide sufficient information about the behavior of an AIES to begin the assessment of the CONEMP envisioned for it. This is an area where models of the AIES can be used in exploration, and the effectiveness of varying CONEMPs can be tested. The goal would be characterization over a variety of designs and CONEMPs. This would initially guide more detailed design. It would also provide a basis for rapid modification of the design, the CONEMP, or both in the event that problems were discovered later.

3.6.2 Characterization of Average versus Outlier Behavior

The correlations between average or typical behavior and worst-case behavior for ML systems are unlike those for physical systems. Optimizing average or peak behavior could lead to worsened outlier behavior. This calls for more complete early characterization of how the ML module behaves under differing inputs. The relative importance of outlier and typical behavior is highly mission specific. T&E professionals, through their awareness of requirements, requirements testing, and COIs, inform decisions about the necessary extent of characterization.

3.6.3 Reward Hacking

Note that all ML systems rely on some form of reward function to refine model parameters. Aligning this reward function with mission success is key to successful development. Any misalignment can lead to development of suboptimal behavior. A human example would be an office that rewarded the number of forms processed by workers—potentially leading to sloppy work and returned forms. Early testing can determine how susceptible the approach is to reward hacking or other suboptimal developments. The impact of design trade-offs to minimize the disconnect between the rewards used in model training and other elements of performance can be explored in cases where reward hacking appears to be a significant risk.

For any specific system development, early attention to the technological elements most likely to exhibit problems late in development can be useful. Certain areas—the connection of design to CONEMP, the relation between average and worst-case behavior, and the prospects of reward hacking—pose likely risk in AIES. Early involvement of T&E professionals in the exploration and characterization of these areas can decrease the probability of problem discovery late in development and mitigate the disruption that such discoveries might cause.

3.7 Summary

Changes in T&E practice are imposed by ML—the data-driven, non-procedural AI that produces models from (usually extensive) datasets. The model and the data are new features of systems, and both require assessment with substantial T&E involvement. Each type of ML has a common high-level structure and development process with four main phases:

- Model Specification
- Data Preparation
- Model Training
- Model Assessment

T&E is instrumental in the success of all these phases.

Testing required for model and data assessment is qualitatively new because the data and types of models are new. In addition, AIES/MLES often impose additional challenges because of the quantitative increase in the operating space and scope of possible actions. This can combine with the brittleness of ML models to substantially increase the test points required for coverage.

M&S and the use of formal methods provide means to mitigate these coverage challenges. M&S can substitute for open-air testing and also in test planning to optimize the information from the conducted tests. Formal methods use mathematics and logic to rigorously evaluate how software will behave. It can be of particular use in early DT in proving that some classes of problems will not occur, obviating the need for some of the testing.

Ensuring the trustworthiness of MLES requires being able to generalize system capabilities and limitations beyond the TVT data used to develop the models. Model brittleness, reproducibility challenges, and the potential for increasing mismatches between TVT coverage and real-world operational environments make this particularly challenging. In addition to black-box testing and formal methods, it can be useful to implement diagnostics that give visibility into the interior workings of the ML models and to continue to monitor these diagnostics during fielded operations. This instrumentation may involve design trade-offs in system power, processing, and cooling requirements; DT&E personnel should discuss the potential costs and benefits of special instrumentation with systems engineers and PMs, beginning as early as possible in the development life cycle.

Early, technology-focused testing can reduce the probability of discovering problems by either DT or OT late in development, with relatively mature, difficult-to-modify designs. The elements of AI that increase the importance of this testing include the interplay between CONEMP and design; the changing correlations between average and outlier behavior of ML systems; and the prospect of reward hacking during model training. Exploration and characterization of these areas can decrease the probability of problem discovery late in development and mitigate the disruption that such discoveries might cause.

4 Expanded Interactions for the T&E Community

4.1 Introduction

The use of AI presents new challenges in providing assurance to the set of stakeholders participating in or impacted by development. T&E has a major role in providing this assurance. Policy mandates such as the DoD AI Ethical Principles (outlined in the May 26, 2021, DepSecDef Memorandum) and DoDI 3000.09 increase both the amount and types of assurance to be provided to a wider set of stakeholders. Optimizing the path from initial concept to fielding and utilization will require convincing these stakeholders that the systems developed are effective and ethical and can benefit from expanded interaction of the T&E community with other groups. The T&E community (with an emphasis on test) can interact with stakeholders early to ensure the timely availability of evidence that stakeholders will require. The T&E community (with an emphasis on evaluation) also has a role in combining the evidence from all sources: traditional tests, M&S, formal methods, and other sources useful to the certifying authorities and other decision makers.

This section describes how T&E professionals can contribute to contracting, requirements development, CONEMP development, and design trade-offs.

4.2 Contracting

The use of AI components requires careful consideration of drivers and constraints that must ultimately be translated as terms and conditions in a negotiated contract mechanism. In particular, the contract will establish the parameters of information sharing among participants. The use of AI presents new challenges to this process. Data rights, including VV&A of training data and data from diagnostic instrumentation, and protection of IP are now potentially important features in the contract. T&E professionals can provide support by advising PMs on the information needed for VV&A of models; estimating the cost and schedule impacts of the failure to obtain specific data rights; and considering the information needed to plan and augment physical testing. By addressing these new challenges, T&E professionals support efficient and effective testing and robust evaluations in support of assurance to stakeholders. (Note that DoDI 8320.02, “Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense,” and DoDI 8320.07, “Implementing the Sharing of Data, Information, and Information Technology (IT) Services in the Department of Defense,” are being updated and combined into a new version of DoDI 8320.02. This update is likely to influence contract specifications for data and other information.)

4.2.1 Access to Contractor Data Improves Test and Evaluation

MLES introduce new dependencies on data for T&E. It is often said of ML that "the intelligence is in the data" and not in the code. As a result, an important part of evaluating an ML model is verifying and validating the TVT data that produced it. The data used to train, validate, and test ML models are key to their development and also to characterization of performance and ultimately evaluation. These data can be government furnished or contractor owned. Increasingly, the development of ML models is expected to use synthetic data in the training of AI models; comparison of AI model training methods; and performance optimization of the AIES. If contractor generated, the synthetic data may be regarded as important IP with originators reluctant to share, leading to more extensive and time-consuming testing.

The Government Accountability Office Report, GAO-23-105850, found that no AI-specific guidance has yet been provided for acquiring major systems with embedded AI. Developing and conducting VV&A will increase the importance ofcontractual provisions enabling access to AI/ML training data and models. Using FAR/DFARS contract provisions, as will likely be the case for major weapon systems, will require special care and attention to obtain the needed access to data and models. The T&E community should be prepared to provide advice in that regard.

For early stages—experimentation and prototyping—a step-by-step approach to contracting, from a program office perspective, can be found in the AI Acquisition Guidebook. This document is useful in providing advice for the early stages of the life cycle.

4.2.2 Verification, Validation, and Accreditation of Training Data

In accordance with DoDI 5000.61, data used to support DoD processes, products, and decisions undergo V&V throughout their life cycles and are accredited for a specific intended use.

(Note that the responsibility for V&V and for accreditation is delegated to the DoD Components. Guidance on contracting policy would reside with the Components.) V&V on the data will continue through the life cycle as data are added or refined, including V&V of synthetic data. This will continue post-fielding. Contracts will establish government visibility into the training data (including labeling and validation thereof), and thus will establish the limits of possible V&V. For government-owned data, this is straightforward. For contractor data and contractor-generated synthetic data, T&E professionals can assist in determining how visibility will affect the ability to perform V&V.

V&V on the use of contractor synthetic data is likely to be especially challenging. New methods may be needed when synthetic data are used in training. Use of synthetic data in the evaluation of ML-produced models is methodologically similar to established practices in using M&S to augment physical testing. The challenge is that ML models and physical systems respond to different features of their input. In some cases, the generation of synthetic TVT data will be faster and easier than obtaining and labeling real-world data, which may change the evolution of the training data in use—putting additional challenges on the need for V&V "throughout the life cycle." T&E professionals can help determine how to assess the overall quality of the data.

Accreditation is for an intended use. To support accreditation, the accrediting authority must examine the data for representativeness of the conditions to be encountered. Within the ML field, this is normally understood as a training set statistically similar to what will be encountered in the field. For many DoD missions' intended uses, however, the system will be required to function well on very rare events, in which case the accreditation must establish success under the most important conditions. In these cases, representativeness will not be sufficient and may be counterproductive. Synthetic data are especially likely to be used in these cases. Accreditation requires an understanding of the mission and the environment, as well as the traditional quality metrics for ML systems. T&E professionals can provide the link between the data used for the building and VV&A of ML models and the requirements and mission needs of final systems. Explicit articulation of this link can support developing contracts that provide access to the data needed for these activities.

4.2.3 Expanded Diagnostic Instrumentation and Data

Diagnostic instrumentation and tools are a key part of developing any complex system. For AI models and AIES, expanded uses of diagnostic information would need to be contracted for in order to be available. This includes instrumentation of decision-making algorithms. Extrapolating the results of tests to other conditions depends on assessing whether decisions are made for the right reasons, not just that the decisions led to acceptable outcomes. Retaining diagnostic instrumentation throughout the system life cycle can be of great value. T&E professionals can advise on the value of retaining the instrumentation in the context of the space, weight, and power claim that the instrumentation will impose.

For applications with significant HMT, the instrumentation of the human may also be essential for characterizing system performance. In particular, measurements of teaming performance under conditions of high workload or high stress for the human are needed. This capability would need to be specified in a contract. Requirements for government provision of "typical end users" might also be a contract feature. At the prototype stage, and beyond, additional instrumentation is needed to diagnose mission-related shortfalls in AIES (Sparrow et al. 2018). The following factors may lead to these shortfalls:

- Insufficiently representative TVT data.
- Flawed decision algorithms.

- Inaccurate information fed to the decision algorithms.
- Flawed integration of the AI into the system in which it is embedded.
- HMI problems tracing to a flawed CONEMP.

Instrumentation of the decision algorithms supports all four focus areas of the CDAO T&E Strategy Frameworks, as discussed in Section 2.3.

T&E professionals can advise on the benefits of expanded instrumentation for the characterization and evaluation functions. In particular, instrumentation that provides insight into whether good (or bad) outcomes traced to good (or bad) decision making or to other causes would have implications for the ability to generalize results from the executed test points and may have implications for the scope of needed testing. This is key for the characterization of the performance envelope as well. This instrumentation will come at a cost to the developer and will need to be addressed in the contract.

4.2.4 Insight into Development Processes Can Contribute to Assurance

A fundamental challenge in the development and T&E of AI, either as a component or as part of a system, is providing sufficient assurance to decision makers and other stakeholders that capabilities will be delivered and bad outcomes will be avoided. The quality of the development processes can be an important part of the evidence underlying any assurance argument. This quality could be part of the selection criteria or contractually addressed in other ways. For the T&E community, understanding the development processes is of value, apart from assessment of the development processes per se.

T&E professionals should be able to advise the contracting effort on the value of insights into the development process. Processes will determine what data on component and system performance, during development and through the life cycle, will be retained. These data, if available, can contribute to evaluations of complete systems. The performance history can provide indications of likely performance in conditions other than those tested, potentially reducing the need for more testing and/or improving the evaluation. This history is also informative about the likely extent of behavior changes as systems are upgraded post-fielding. Understanding what information on the development process will be available is an asset to T&E from initial planning through final assessments and post-fielding sustainment. The developer’s handling of performance data impacts T&E execution and thus impacts the program. This can be anticipated and incorporated in contracting efforts.

Section 3.2.2.3 addressed VV&A of training data. The discussion was aimed at assessing the quality of the data itself. The nature (and quality) of the process to produce the data can be informative as well. For SL systems, the process for segregating data into TVT components is important for system robustness. For RL, the process of determining an objective function that captures mission-essential performance is similarly important. How synthetic data might be produced and used in training for SL and RL systems can be informative to evaluators. It can also be sensitive to contractors. Obtaining information about these processes would need to be a negotiated contract provision.

4.2.5 Protection of Contractor Intellectual Property

Contracting for information about the developer’s processes may raise issues of protecting the contractor’s IP or proprietary information. The details of ML models produced may also be regarded as IP and in need of protection. T&E professionals can contribute to the contracting negotiations by articulating the value that sensitive information has in terms of its impact on the cost and schedule of testing and on the validity of the resulting evaluation. The value of any information and its sensitivity will be highly dependent on the specific development in question, so generalizations are difficult. In some cases, it might be worthwhile to solicit bids with varying levels of contractor data and IP sharing. T&E professionals could comment on the “value” in terms of more rapid or less expensive testing associated with different levels of information sharing.

Notwithstanding the highly system-specific nature of these value propositions, early engagement by T&E professionals can be of value. In particular, influencing the early communications in the form of “Sources Sought” or “Request for Information” documents can shape expectations and allow for development of an optimal path. This approach exemplifies both “shift left” and “shift right” strategies—considering the impact of developer openness on the effort required to build the assurance needed for successful fielding well before a program of record exists.

4.2.6 Specific T&E Community Contributions

The essential contribution of the T&E community to the contracting process will be articulating the value to the program of contracting for specific information. This may be information about the training data; how the data were used; available performance history during model development; or other aspects of the system. Such information may be of use in test planning; in reducing the scope or increasing the value of tests that are executed; or in improving the quality of evaluations. In some cases, the contracting will be a straightforward reimbursement for the effort of collection and retention of data. In other cases, the information will be regarded as IP, of value far beyond the cost of handling. Appropriate contracting for information requires clarity on the value that the information will carry for the program office and other stakeholders.

4.3 Requirements Development

This section discusses "Requirements" in the "Big R"-related sense—that is, Joint Capabilities Integration and Development System (JCIDS) requirements. Use of AI is a design choice and should not be an explicit capability requirement. The T&E role is in monitoring design and determining and reporting testing implications for *how* a requirement is to be met. Inclusion of AI in the capability solution introduces additional elements to consider in requirements formulation:

- T&E has traditionally ensured requirements "testability." This is necessary for characterizing the system (or component) under test. For AI components or AIES, characterization is additionally challenging because of AI fragility and sensitivity to small changes in conditions. This can drive a need for extensive testing. The T&E role thus expands to ensure the feasibility and affordability of system characterization needed to provide assurance.
- Modern software development emphasizes flexibility in the satisfaction of technical requirements, as part of a highly iterative design process. Preserving satisfaction of the KPPs and KSAs while tolerating iteratively changing technical specifications is an additional challenge.
- AI systems will alter the process of requirements refinement during development. More complete characterization of performance will allow for trades between threshold shortfalls and objective exceedance that preserve mission success—despite not meeting initial threshold performance.

4.3.1 Mandatory Requirements

The Manual for the Operation of the JCIDS lists four mandatory KPPs and two mandatory attributes:

- Force Protection KPP
- System Survivability KPP
- Sustainment KPP
- Energy KPP
- Interoperability Attribute
- Exportability Attribute

4.3.1.3 Force Protection Key Performance Parameter

The JCIDS Manual states that the Force Protection KPP is intended to ensure protection of occupants, users, or other personnel who may be adversely affected by the system or threats to the system.

The presence of AI in a system is unlikely to have a significant impact on requirements flow-down or testing related to the Force Protection KPP. Changes in force protection requirements may be driven by changes in threat. Changes due to design iterations or trade-offs with other requirements rarely occur after the earliest stages of a program.

4.3.1.4 System Survivability Key Performance Parameter

The JCIDS Manual states that the System Survivability KPP is intended to promote the development of critical warfighter capabilities that can survive kinetic (i.e., traditional; nontraditional; and chemical, biological, radiological, and nuclear (including electromagnetic pulse)) and non-kinetic (cyber and electromagnetic spectrum) threats across domains and applicable environments including space.

The presence of AI in a system may introduce new vulnerabilities, primarily in the cyber domain. The testing needed to address cyber challenges to system survivability may depend upon how the AI features in the system contribute to meeting the entire set of system requirements. The forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook will provide additional information.

4.3.1.5 Sustainment Key Performance Parameter

The JCIDS Manual states that the Sustainment KPP is intended to ensure an adequate quantity of the capability solution will be ready for tasking to support operational missions.

The presence of AI in the materiel solution is unlikely to affect “sustainment” as defined above, which is driven by reliability, maintainability, inventory, etc. However, “sustainment” in the sense of “keeping fielded AIES working” requires monitoring and regression testing, which could be substantial.

4.3.1.6 Energy Key Performance Parameter

The JCIDS Manual states that the Energy KPP is intended to ensure combat capability of the force by balancing the energy performance of systems and the provisioning of energy to sustain systems/forces required by the operational commander under applicable threat environments.

The mandatory Energy KPP is aimed at theater-scale use of energy. The presence of AI in systems is unlikely to impact energy use on this scale. Energy demands by AI components of AIES may be relevant to the capability solution being developed, but the testing issues will be common to any energy-requiring components.

4.3.1.7 Interoperability Attribute

The JCIDS Manual states that this attribute addresses Physical Interoperability, Net-Ready Interoperability, and Joint Training Technical Interoperability in a Source Table. The performance attribute ensures interoperability between individually developed and fielded capability solutions.

Ordinarily, interoperability is tested as “information technology (IT) interoperability” and depends primarily upon interfaces. This would be largely unaffected by the presence of AI within a system. An exception would be if AI features were guiding or embedded in the interface operation. If AI is involved in mediating or controlling the content or timing of critical information exchanges, additional assurance will be required to establish interoperability. For either situation, additional developmental and certification testing may be required.

Note that the definition of interoperability in DoD is significantly broader than IT interoperability. These attributes would be encompassed by the KPPs, KSAs, and additional performance attributes (APAs) associated with the capability solution. The presence of AI “embedded” in a system, but not controlling or influencing interfaces, could still affect the broader interoperability of the capability solution. System integration testing may be needed to confirm the compatibility of AI components with other system elements. Testing to determine dependencies between the system under test and external systems may be influenced by the presence of AI and require specific testing to determine the impact on these dependencies.

4.3.1.8 Exportability Attribute

The JCIDS Manual states that a KSA will address exportability for the program, in accordance with the April 15, 2019, VCJCS Memorandum, “Conventional Arms Transfer Policy, National Security Presidential Memorandum Task 2.7, Build Exportability.”

As defined in the JCIDS Manual, exportability is the process to identify, develop, and integrate technology protection features into U.S. defense systems early in the acquisition process to protect Critical Program Information and other critical technologies/capabilities and thus enables a system’s export to partners. Technology protection primarily involves two tools: anti-tamper and differential capability modifications.

Utilization of these tools is unlikely to be affected by the presence of AI.

4.3.2 KPPs, KSAs, and APAs Essential to the Capability Solution

Most of the attention to the interplay of testing AI elements or an AIES and the requirements for that system will manifest themselves in the requirements specific to the capability solution. It will be challenging to capture capability requirements that can be tested in ways that are both possible in principle and feasible in practice. These challenges are driven by the potential for small changes in conditions to lead to large changes in results and by the variability of humans when interacting with complex AIES.

In general, comparative requirements, such as “detects targets as well as humans” or “maneuvers as well as a legacy system” are difficult to establish. Comparative requirements are prone to ambiguities such as the following: As good at peak behavior or on average? Less likely to perform badly? Better than the best human or the average human?

The inclusion of AI features aggravates the problems with comparative metrics. Human baselines are ill-defined—and especially variable for any tasks that are likely to be AI related. The machines are not variable in the same way as humans, but the sensitivity to small changes in the environment creates similar problems in assessing any comparative requirements.

Emerging regulations also lead to de facto requirements. These de facto requirements are likely to become incorporated in formal KPPs, KSAs, and APAs in much the same way as cybersecurity has been incorporated. At present, it may be useful to address how to satisfy them in the standard requirements process. These include AWS, modular open systems architectures, and the DoD AI Ethical Principles. In accordance with DoDD 3000.09, autonomous and semi-autonomous weapon systems will be designed to allow commanders and operators to exercise appropriate levels of human judgment over the use of force. This requirement on the design will need to be specified as a testable hypothesis.

Confirming compliance with Modular Open Systems Approach standards will primarily depend upon inspection and analysis. However, technical aspects to the interfaces between modules must be satisfied in an open architecture. These will have bandwidth, latency, and other features needed to ensure the advantages of the open architecture. Expressing these requirements in testable form will benefit from T&E personnel participation.

The DoD AI Ethical Principles require AIES to be:

- Responsible
- Traceable
- Reliable
- Governable

Expressing these high-level principles as testable quantities will be challenging. Quantifiable metrics are not well established for these principles.

The T&E community can play a role in how to quantify important system requirements in ways that can be tested effectively and efficiently. AIES are likely to warrant additional attention to testability issues, particularly for the capability (as opposed to mandatory) requirements. Requirements that derive from a policy mandate as well as desired system capabilities also warrant attention.

4.4 Concept of Employment

4.4.1 Introduction

System design restricts possible CONEMPs for all systems. For MLES, these constraints can become more complex as machines become less like tools and more like teammates. In addition, the iterative nature of design for MLES can require iterations in the CONEMP to maintain or optimize performance. Testing during design can characterize the impact of design iterations of the ML model on the effectiveness of the envisioned employment of the MLES. This can be an important element in optimizing performance of the final system. Interaction of the T&E community with requirements developers and the program office for the system early in CONEMP development is potentially of value.

4.4.2 Human-Systems Integration Test and Evaluation

Any CONEMP for an AIES will depend upon the information flow between the machine and the human. What is communicated, when it is communicated, and how it is communicated will all influence system employment. The basic machine features can be tested straightforwardly. The effectiveness of associated CONEMPs will, in general, require testing (and possibly dedicated infrastructure) to determine the effectiveness of the human-machine combination.

One approach to ensure safety and preclude unacceptable behavior by AI-enabled machines is to incorporate “runtime monitors” (Section 3.5.3.4). If the runtime monitor function is performed by a human, latency and timeliness of information will need to be measured to assess the effectiveness of monitoring. Monitoring must remain effective under high levels of human workload and fatigue and also under conditions of boredom. T&E professionals can advise program officials and user representatives about what to measure and how to measure it to facilitate coherence between design and CONEMP as both are iteratively developed.

This discussion of HSI T&E also applies to AWS, which has explicit requirements on the effectiveness of human control.

4.4.3 Calibrated Trust

Any CONEMP for using AIES will depend upon the user having properly calibrated trust in the AIES. Several elements of the design and CONEMP must be addressed to achieve calibrated trust.

Humans exhibit great variability (both between different humans and for a given human over time) in how they respond to AIES. Well-studied phenomena exist in automation bias—the tendency to over-trust—and in algorithm aversion bias—the tendency to under-trust (Goddard et al. 2012; Jussupow et al. 2020; DeCostanza et al. 2018); see also Section 2.2.3.4. Innate variability can be addressed with the training of personnel or with the restriction of personnel who can be selected as users (or both), as part of the CONEMP. How users would be chosen to maximize dependability of the human-machine system will depend upon the specific design chosen, in ways that can be tested. For example, different designs might exhibit greater or lesser variability in the responses of different users. Service S&T organizations all have substantial expertise in this area.

4.4.4 Emergent Behavior

Emergent behavior refers to behavior of a system that is not (or is not in any obvious way) predictable based on an understanding of the system components (Trusilo 2023). Regarding an AIES together with its human user as a system, emergent behavior then refers to unexpected aspects of the interaction between the two. How the human-machine integration is executed can have a significant effect on the frequency and severity of emergent behavior. Further, adverse emergent behavior may overwhelm experiences that may have built properly calibrated trust. Designing an AIES and developing an associated CONEMP that avoids bad outcomes of emergent behavior, which is by definition difficult to predict, will be challenging for the foreseeable future. In this context, T&E professionals can identify procedures to characterize emergent behavior when it occurs and retain the data on this behavior for future use throughout development. An extensive track record will be informative for ongoing refinements to design or the CONEMP or both. The track record also may be important in developing the assurance package for the system.

4.4.5 Human-Machine Interactions

HMIs are increasingly important with the advent of AIES and come into play even in the absence of emergent behavior. Established metrics exist for the quality of these interactions, which allow for a more nuanced assessment of the design/CONEMP synergy than outcome-based observations alone. Research in this area is increasingly focusing on the interaction elements. Candidate metrics address shared perspective, cooperative behavior, and resource/tasking allocation between the machine and the human (see Sections 2.2.3.4 and 2.4.4.1). These metrics can be used, in conjunction with traditional mission performance metrics, to provide insight into how the design/CONEMPs synergy will change with different circumstances. The T&E community can advise on what features inform the iterative development of design and CONEMPs and how to measure those features.

4.4.6 DoD AI Ethical Principles

The CONEMP for any DoD system will be required to abide by the DoD AI Ethical Principles and related issuances. These principles were first officially stated in the May 26, 2021, DepSecDef Memorandum. They have been reaffirmed, and nearly all DoD (and Federal) policies are captured in this early formulation:

- **Responsible**: DoD personnel will exercise appropriate levels of judgment and care, while remaining responsible for the development, deployment, and use of AI capabilities.
- **Traceable**: The Department's AI capabilities will be developed and deployed such that relevant personnel possess an appropriate understanding of technology, development processes, and operational methods applicable to AI capabilities, including transparent and auditable methodologies, data sources, and design procedure and documentation.
- **Reliable**: The Department's AI capabilities will have explicit, well-defined uses, and the safety, security, and effectiveness of such capabilities will be subject to testing and assurance within those defined uses across AI capabilities' entire life cycle.
- **Governable**: The Department will design and engineer AI capabilities to fulfill their intended functions while possessing the ability to detect and avoid unintended consequences, and the ability to disengage or deactivate deployed systems that demonstrate unintended behavior.

With regard to the first principle, "responsibility" in DT&E involves activities to support assurance of the other four RAI principles. Those principles have implications for DT&E as follows:

- **Traceable**: The requirement that "relevant personnel possess an appropriate understanding of technology, development processes, and operational methods" implies a link between CONEMPs and design. Different CONEMPs will require different levels of understanding by the operational users of the underlying technology and development processes.
- **Reliable**: This principle is essentially a mandate for AIES to have a CONEMP with "explicit, well-defined uses" subject to test for "safety, security, and effectiveness." Note

that the exact wording is “testing and assurance.” This includes the evaluation role in T&E, where evidence of compliance from sources other than test are included as part of an assurance package.

- **Governable**: Ensuring that the CONEMP for a system satisfies the “governable” principle is primarily a human-machine integration testing activity. Addressing information flow and the practicality of human control under realistic operational conditions are the essential elements in establishing governability and are readily addressed by a properly designed test regime.

In these four areas, interactions between CONEMP developers, system designers, and T&E personnel can clarify how the system/CONEMPs combination can be shown to be consistent with the DoD AI Ethical Principles.

The iterative nature of ML-model development can lead to an iterative evolution of the system CONEMP as well. This evolution can be informed by a program of HMT that can be developed and executed by the T&E community, in conjunction with the system designers and CONEMP developers. In many cases, this will be the optimal path to effectiveness of the system as employed.

4.5 Design Trade-offs

T&E of AI components and AIES has been recognized as posing additional challenges for roughly a decade or more. In addition, expanded mandates derive from the DoD AI Ethical Principles and the DoD Responsible AI Strategy and Implementation Pathway, as well as DoDD 3000.09 for AWS. Addressing these challenges and providing assurance to the expanded set of stakeholders increases the scope of the T&E and related activities. This increased scope of T&E and related evidence-generating activities such as the use of M&S and formal methods may introduce new elements in the design trade-space. The impact on design is only one element of a broader set of implications for the systems engineering of AIES imposed by the need for successful T&E. The MITRE Technical Report, “Systems Engineering Processes to Test AI Right (SEPTAR) Release 1,” (Balhana et al. 2023) calls for evaluation of the entire systems engineering life cycle to characterize AIES trustworthiness, from requirements through sustainment.

Different design approaches will require different test approaches—and often different test infrastructure. T&E can contribute to the design decisions by articulating what is necessary to provide assurance for a given approach. The associated cost and schedule for developing, testing, and obtaining approvals may vary significantly for different designs. T&E professionals are well-versed in planning for the cost and schedule impacts of different testing regimes.

4.5.1 AI Type and Architecture Choice

4.5.1.1 AI Type

The initial decision is whether to use an ML approach or use procedural AI or traditional software. In general, ML approaches may be expected to require more extensive testing. The sensitivity of ML to small changes in input data generally requires a denser sampling of the input space. Furthermore, integration of ML capabilities within a system often involves more complex coupling of the ML with the system than more traditional methods. The result is more frequent iteration of the ML model design itself and more frequent design iteration of the overall system. The iteration needs to be supported by model, human-system interaction, and system integration testing. The impact of sampling and testing to support these iterations can be estimated by T&E personnel with experience in test planning. This becomes a design trade issue when the design expected to produce the best performance has significantly increased cost and schedule burdens.

Missions may exist where multiple ML approaches are competing candidates for providing a particular capability. Within the set of ML approaches, how to test for performance and ensure dependability will differ between UL, SL, and RL. This is another area in which T&E impacts the performance, cost, and schedule trade-space.

For all the ML approaches, or comparison of ML with procedural approaches, the availability of TVT data is a potential cost and schedule driver. This includes the need for VV&A on any data that are to be used. In some cases, the T&E community will be called upon to execute the VV&A, either as part of the test team within a program, part of Service operational test agency responsibilities, or as an OSD function.

4.5.1.2 System Architecture

The system architecture can also have implications for the burden of testing necessary to demonstrate performance—but especially to provide assurance of safe and dependable behavior. One commonly advocated approach is the use of a runtime monitor, or similar system, to provide RTA that bad system behaviors will not occur. In one sense, this is already a trade-off between assurance and performance, although a dynamic rather than static trade-off. The usual presumption is that assurance cannot be provided for an ML system that is typically very high performing but may have occasional unacceptably bad behavior.

The architecture choice then is to embed the high-performing ML with another, lower-performing component, known to be safe, with an overseer that will switch between them as needed. This introduces new testing requirements for the "known to be safe" component, the overseer, and the interfaces between them. The considerations are essentially the trade-offs between a difficult-to-test system of systems versus a difficult-to-test stand-alone system. Runtime monitors are a specific example of this approach, as discussed in Section 3.5.3.4.

4.5.2 Applicability of Non-Test Evidence for System Behavior

The testing of AIES can be augmented by the use of M&S and formal methods, as discussed in Sections 3.3 and 3.4, respectively. In general, these techniques are deployed when it is difficult or impossible to accumulate the evidence to provide an assurance package with physical testing alone. These techniques come with costs themselves, and these costs may be expected to vary depending upon the design approach taken.

Models that have been developed to drive traditional software systems, even if they have been verified, validated, and accredited for that use, will require additional testing to establish validity and accreditation to drive ML systems. This is because ML systems are sensitive to high-dimensional and possibly high-frequency features of the training data that are ignored by humans, analog systems, or traditional software. The VV&A for the new application can be expensive and time-consuming and may not succeed. Development of a dedicated M&S driver of the ML components, if necessary, would potentially be time-consuming and expensive.

Formal methods are likely to be more applicable and more easily applied to procedural AI than to ML approaches. Applying formal methods to ML approaches remains largely a research area. Applying formal methods to procedural AI can exploit the extensive work done on formal methods applied to software more generally.

Note that under some circumstances, the processes followed in developing the AI capability might have a role in providing assurance to at least a subset of stakeholders. This would be very dependent on the details of the program and the historical record of the participants.

The use of AI presents new challenges in providing assurance to the set of stakeholders participating in or impacted by a development. The burdens these challenges may impose on program execution in terms of cost and schedule are likely to depend upon specifics of the chosen AI design. Under some circumstances, it is appropriate to include the burden of providing assurance in the design trade-space. T&E professionals have a major role in providing this assurance and are uniquely qualified to forecast the burdens and participate in the trade-space.

4.6 Accreditation and Certification Support

In accordance with forthcoming DoDI 5000.DT, the PM charters an integrated test planning group to develop a strategy for robust, efficient testing to support certifications throughout the acquisition life cycle. DT&E traditionally provides important data in support of a variety of certification requirements administered by multiple authorities. The key certifications include safety; cybersecurity and software assurance; joint interoperability; and platform- or mission-specific certifications (such as airworthiness or DoDD 3000.09 approval). Emerging Federal government and DoD policy is also moving toward establishing an equivalent of certification for adherence to RAI principles. Certification of AIES can be especially challenging because of the interaction of AI-specific risks and uncertainties (see Section 2.2.2) with the certification process.

In addition, as noted in Section 3.2.5, DoDI 5000.61 calls for VV&A of models used in DoD processes and systems as well as VV&A of the data used to develop them. DT&E has traditionally been closely involved in VV&A of simulation models and can expect to be similarly involved in VV&A of ML models and the data used to develop them in the future. The dynamic nature of ML models, with frequent updates and retraining as additional data are collected, places a specific burden on accreditation, which includes a judgment of which new applications or revised models require revalidation and which do not.

Although the precise impact of AI on certification and accreditation support will depend on the type of AI used and the mission, certain particular features of MLES are most likely to impact certifications. These include quality of TVT data; HMIs, especially if significant teaming occurs; ML model brittleness; model robustness; vulnerability to adversarial actions; and model development pipeline dependencies. The following subsections discuss the potential certification impacts of each of these features.

4.6.1 Quality of TVT Data

Data quality is the fundamental enabler of ML. Assessment of quality is obviously the main focus of data VV&A activities, but data quality also potentially affects nearly all certification activities. RAI evaluations need to assess any potential privacy or unwanted bias issues in the available datasets. Safety engineering and airworthiness/seaworthiness certifications need to verify that any potential safety hazards traceable to insufficient coverage or representativeness of the TVT data have been adequately mitigated. Cyber authority to operate will depend on both data security and avoidance of data poisoning. For AWS, DoDD 3000.09 review will need to establish that overrepesented or underrepresented entities or environments in the TVT data do not create an unacceptable risk of unintended engagements. DT&E activities and outputs can support all these certification needs.

4.6.2 Human-Machine Interactions

Most certifications include usability criteria, explicitly or implicitly. For RAI, the governability of AIES depends on the ability of humans to exercise effective control of systems in a practical chain of command. For safety engineering, human factors such as induced workload, operator training requirements, fatigue, and required human response times for safety-critical actions must be assessed. For AWS, suitable HMIs may be necessary for avoiding unintended engagements and maintaining effective control of systems. DT&E activities and outputs can improve data collection in support of the required assurance arguments for these concerns. See Section 2.2.3.4 for additional discussion of DT&E related to HMIs.

4.6.3 Model Brittleness

The potential for model brittleness, as described in Section 2.2.2, is a key concern of model VV&A. Characterizing the sensitivity of models to small changes in input and identifying problematic regions of the input space are important activities supporting engineering of the MLES and characterization of system capabilities, limitations, and risks. This characterization is also valuable to testers for understanding the limitations of test reproducibility and how to interpret variable system behavior under repeated trials.

Model brittleness also directly affects some certification activities. The RAI requirement that systems be reliable (as noted in Section 2.2.2.1) calls for identifying and mitigating model brittleness, as do safety and airworthiness certifications. For AWS, model brittleness is a potential source of unintended engagements to be accounted for in the approval process.

4.6.4 Model Robustness

Model robustness, as defined in Section 2.2.2.2, refers to model worst-case performance, as opposed to average or peak performance. Robustness is a key aspect of model VV&A and also of data VV&A to the extent that failures of model robustness can sometimes be traced to data issues. The need to separately characterize the nature and likelihood of worst-case model performance, and the system-level mitigations in place to handle those cases, is clearly relevant to safety, cyber, airworthiness, joint interoperability, and DoDD 3000.09 evaluations. It is also relevant to assurance of RAI precepts, in that system robustness directly impacts governability and reliability. DT&E activities and outputs can help inform certification authorities of limitations and risks associated with model robustness.

4.6.5 Vulnerability to Adversarial Actions

Data VV&A includes assessing TVT data for potential data poisoning. Model VV&A includes identification and characterization of any vulnerability to adversarial inputs, either through cyberattack (insider or external) on the model or through environmental manipulation. DT&E activities and outputs can inform both of these processes.

Adversarial vulnerabilities must be accounted for in RAI evaluations, with regard to the governability of the fielded system. For safety and airworthiness, assessment of mitigations to safety hazards must account for potential adversarial vulnerabilities of any models with safety-critical roles. AWS reviews must account for adversarial vulnerabilities in assessing the risk of unintended engagements or loss of effective control. The forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook will discus cyber T&E implications of potential adversarial actions.

4.6.6 Development Pipelines

The ML development process shown in Section 3.2 has implications for accreditations and certifications. Data VV&A must consider the processes used for:

- Data collection, conditioning, transformation, and normalization.
- Protection of privacy and avoidance of unwanted bias.
- Data augmentation, including missing value imputation and synthetic data.
- Separation of TVT instances into appropriately representative training, validation, test, and independent test datasets.

Model VV&A should evaluate:

- The cross-validation strategy and other techniques for avoiding overfitting.
- Alignment of the training reward function with mission priorities.
- Techniques used to enhance robustness or reduce adversarial vulnerabilities.

In support of the RAI traceability principle, evaluators will need to understand data provenance and the model development process, including any use of XAI techniques. Software assurance will need to be aware of any supply chain issues associated with data, models, or tools used in their collection and processing.

4.6.7 Certification – ML Crosswalk

Table 4-1 summarizes the dependencies described in this section. Data and model VV&A activities must account for ML-specific challenges. Furthermore, many certification activities are affected by the use of ML and may be informed by the outputs of DT&E. A more detailed description of DoD safety engineering policy and practice is provided by Military Standard MIL-STD-882E, “System Safety.” Cyber T&E of AIES will be addressed in more detail in the forthcoming Version 3.0 of the DoD Cyber DT&E Guidebook. Joint interoperability testing processes and requirements for certification are described in Version 3.0 of the DoD

Interoperability Process Guide. RAI requirements and guidance can be found in the DoD Responsible AI Strategy and Implementation Pathway.

Table 4-1: Crosswalk of Certification Needs and ML Issues

	RAI	Safety	Cybersecurity	Interoperability	DoDD 3000.09	Airworthiness	Data VV&A	Model VV&A
Quality of TVT Data	bias, privacy	coverage, representativeness	data poisoning, data security		unintended engagements	coverage, representativeness	bias, privacy, coverage, representativeness, etc.	
HMT	governability	CONEMP		CONEMP, human factors	loss of control, unintended engagements	CONEMP		operator trust, explainability
Brittleness	reliability	predictability			unintended engagements	predictability		reproducibility
Robustness	reliability	worst-case behavior	worst-case behavior	emergent behavior	unintended engagements, loss of control	worst-case behavior	root causes of robustness failure	worst-case behavior
Vulnerability	governability	risk mitigation	adversarial inputs		loss of control	risk mitigation		
Model Development Pipeline	traceability		supply chain				synthetic data, normalization, etc.	objective function alignment

4.7 Summary

For AI components and AIES/MLES, DT&E must adapt to support the inherently iterative nature of ML development. Successful development from concept through fielding and utilization will put a premium on interactions across bureaucratic and community boundaries. This section considered some ways in which the DT&E community can improve the likelihood of successful and timely development through enhanced interactions with other communities.

In traditional acquisition, DT&E activities have not routinely informed contracting. For MLES, however, DT&E can provide contracting officers with important information on how the nature and extent of data rights acquired by the government might affect both development and T&E costs and schedules. Traditional data rights might not routinely provide government access to key information about ML components or MLES—most notably information about TVT data and model features. Access to the appropriate data and models could enhance both efficiency of test execution and test support to development. Where appropriate, the DT&E community could also advise on the value of vendor information specific to ML.

DT&E does have a traditional role in interacting with the requirements community to ensure that system requirements are testable. The challenges of testing MLES broadens this concern to include ensuring that the requirements are not merely testable in principle but that a viable test program supporting the needed evaluations is feasible as well.

The iterative nature of ML development, and the close coupling of system design with the CONEMP, drives a need for T&E measurement activities that inform both system developers and CONEMP developers. In particular, testing in the areas of HSI, calibrated trust, emergent behavior, HMT, and adherence to RAI policies will be needed to avoid costly and time-consuming rework in both system design and CONEMP.

The use of ML potentially expands the scope of DT&E activities. Choices of model type and architecture not only affect test planning but also introduce a potential need for other kinds of evidence to support evaluations. These might include use of formal methods, use of AI-enabled red-teaming techniques, and process-based analyses. Introducing these new activities not only can affect the cost and schedule of a viable test program but also may require interactions between testers and other technical communities.

In some cases, substantial differences will exist in the cost and schedule impact of the different test requirements resulting from different AI approaches. In these cases, T&E professionals can interact with system designers and provide valuable insight into the cost/schedule/performance trades.

This section also noted that multiple certifications are needed by programs at various stages in development and that the use of ML can potentially have a significant impact on what is needed to achieve these certifications. T&E professionals have a role in identifying AI-specific testing or measurements that may be needed to support certifications.

Glossary

accreditation. The official certification that a model, simulation, or distributed simulation is acceptable for use for a specific purpose (DoDI 5000.61). Accreditation of TVT data involves assessing the range of applications and environments for which the data can support a valid ML model.

accuracy. As a generic term, refers to the ability of a model to mimic reality. In the context of ML, refers to how closely model predictions on test data match ground truth. For classifiers, accuracy is computed as the proportion of test outputs that are either TP or TN. For regression models, accuracy is computed as a function of the residuals, such as MSE or MAE.

adversarial instances. In ML, the inputs to a model that are deliberately crafted to fool the trained model. The term is most common in computer vision applications to refer to appliqués and camouflage that are not confusing to human vision but confound trained image recognition models. The practice of using adversarial instances is sometimes called adversarial AI.

anomaly detection. An application of UL to identify unusual or potentially harmful departures from ordinary input data.

artificial intelligence (AI). This guidebook does not attempt a rigorous definition of AI. For the purposes of T&E, the systems that pose new challenges are those featuring computations that are opaque to users, inherently unpredictable, sensitive to small changes in input, and/or dependent on training data and learning algorithms.

automation bias. A psychological tendency in some humans to consistently overestimate or underestimate the dependability of automated systems. Automation bias can be a confounding factor in testing suitability in human-AI teaming situations.

autonomous capabilities. Autonomy, or autonomous capabilities, refers to the ability of a system to make certain decisions or take certain actions without immediate human direction or control. Autonomous capabilities can be system requirements; how they are implemented is typically a design decision.

autonomous weapon system(s) (AWS). A weapon system that, once activated, can select and engage targets without further intervention by an operator. This includes, but is not limited to, operator-supervised AWS that are designed to allow human operators to override operation of the weapon system, but can select and engage targets without further operator input after activation. (DoDD 3000.09)

bias. See “unwanted bias.”

classification/classifier. Classification is the task of determining which elements of a set of predefined category labels apply to a specific input data instance. Models that perform classification are called classifiers.

clustering. A UL task that seeks to assign data instances to groups such that instances within each group are more similar than instances in different groups.

concept of employment (CONEMP). A description in broad terms of the application of specific technologies, processes, weapons systems, or forces to perform a particular mission, task, or procedure. CONEMPs are the most specific of all military concepts and contain a level of detail sufficient to inform the establishment of programmatic requirements. (Chairman of the Joint Chiefs of Staff Instruction 3030.01A)

confusion matrix. For a classifier, a 2x2 array showing the relative frequency of TP, FP, TN, and FN outcomes. These can be expressed as raw counts or as rates.

coverage. See "data coverage."

cross-validation. The process of systematically varying which identically distributed subsets of the available data instances are used for training and which are used for validation during the learning phase of developing an ML model.

data card(s)/data sheet(s). Structured summaries of essential facts about various aspects of ML datasets needed by stakeholders across a project's life cycle for RAI development (The Data Cards Playbook Website). They include administrative information and technical information about how the dataset was created and used.

data coverage. In ML, a data quality attribute related to the sufficiency of the TVT data instances to support the training of a model to do the desired task. Good coverage requires having a sufficient number of all of the cases the model is intended to be able to handle with sufficient variety for the model to be able to learn and generalize.

data instance. In SL and UL, one point from the TVT dataset in the representation that will be used as operational inputs to the model.

data labels. In SL, data labels provide ground truth information about the correct model outputs for TVT data instances used as inputs to the model. Models are trained, validated, and tested based on how well they correctly predict the labels of TVT instances.

data poisoning. The deliberate corruption of TVT data in a way that may be overlooked during model development but is expected to adversely affect operational effectiveness.

data sheet. See “data card.”

distance metric. In clustering, a nonnegative numerical measure of the similarity of two data instances with smaller values indicating more similarity. Identical instances have a distance of zero.

emergent behavior. Coherent behavior at the system (or system of systems) level that is not readily predictable from an understanding of behavior at the component level. The tendency of people to speak simultaneously and then be silent simultaneously in lagged communications is an example of emergent behavior in a human-machine system.

expert systems. A procedural AI technique that uses branching if-then logic to recommend courses of action in a variety of complex situations.

explainable AI (XAI). A blanket term for various methods intended to improve user and stakeholder understanding of the rationale(s) behind ML model outputs.

F_1 score. A widely used measure of classifier performance computed as the harmonic mean of precision and recall.

false negative (FN). For a classifier, an FN is when the classifier says that a given label does not apply to an input, and this is incorrect.

false positive (FP). For a classifier, an FP is when the classifier says that a given label applies to an input, and this is incorrect.

Fowlkes-Mallows (F-M) index. A widely used measure of classifier performance computed as the geometric mean of precision and recall.

generative AI (GenAI, GAI). A generic term for any AI system used to generate content such as text, imagery, computer code, or other modalities.

ground truth. In SL, refers to the labels associated with a data instance in the TVT dataset.

heuristic optimization. Heuristic optimization algorithms attempt to optimize mathematical functions for which exact solution procedures are either unknown or too computationally expensive to be used. Large ML models are trained using heuristic optimization of a learning objective function.

hyperparameters. In ML, the hyperparameters of the learning algorithm control how the heuristic optimization uses validation results to adjust the parameters of the model being learned. Hyperparameters are not part of the deployed ML model.

imputation. In data science, a procedure for reconstructing missing values using information from known fully specified data instances. For example, in a database of heights and weights, a missing weight might be imputed using the average weight of people in the database with similar height.

independent test dataset(s). Subsets of the TVT dataset that are set aside for independent testing of the trained model to be deployed, possibly augmented using synthetic data. None of the instances in an independent test dataset should have been used in the training of the model.

instance. See “data instance.”

label. See “data labels.”

labeled data. See “data labels.”

large language model(s) (LLM). GenAI models that use deep-learning algorithms and are pre-trained on extremely large textual datasets that can be multiple terabytes in size. Most LLMs produce text in response to natural language prompts. LLMs can be used to summarize bodies of text, produce prose or poetry, write computer code, engage in interactive dialog, and perform other language-based tasks.

learning algorithm. In ML, the learning algorithm determines how the ML model adjusts model parameters in response to validation outcomes to improve model performance.

machine learning (ML). An approach to algorithm development that, rather than specifying logical or mathematical procedures to be used to solve a particular kind of task, instead learns those procedures from large numbers of examples and encodes them in a flexible parametric model.

Markov decision process. A mathematical model of discrete-time process control in which decisions are made based on the state of the current system and the next state is a function of both the decision made and outside processes (including randomness). Most RL systems model the policy to be learned as a Markov decision process.

model. A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process (DoDI 5000.61). In the context of ML, a model is a parametric function that is optimized to perform well at a particular task, learning from large amounts of data and iterative training.

model card. A short document that provides key information about an ML model. Model cards increase transparency by communicating information about trained models to broad audiences.

neural network. Artificial neural network models consist of layers of nodes, called neurons, that are highly interconnected between adjacent layers. There is an input layer that takes in the data to be processed by the model and an output layer that produces the model's result. In between, there are "hidden layers" of nodes that take inputs from the previous layer and send outputs to the next layer. Each node has an associated weight and an activation function that decides what to pass to the next layer, given its current inputs. The network is trained by adjusting the weights.

normal distribution. In probability and statistics, the normal distribution (also called the Gaussian distribution) is a symmetric, continuous probability distribution with a familiar bell-shaped curve. It is important and widely used, in part because it is the large-sample limiting distribution of many statistics and processes.

objective function. In the context of ML, the measure of the quality of solutions that the learning process is trying to optimize. For SL, this will be some function of the residuals (for regression) or confusion matrix (for classification). For RL, it is a reward function.

overfitting. In the context of ML, overfitting occurs when the parametric model misinterprets random noise or accidental correlations in the data as a pattern to be learned. Overfit models do not generalize well to new inputs. The validation step in the iterative learning algorithm is intended, in part, to detect and correct overfitting.

parametric model. A function with many user-tunable coefficients such that many different relationships between inputs and outputs can be represented through different combinations of coefficient values. In ML, many different kinds of parametric models are used, including artificial neural networks, support vector machines, and Markov decision processes.

policy. In the context of RL, a policy or strategy is a parametric representation of how the RL model will respond to being in a particular state and receiving a particular input.

precision. In SL, the dependability of positive outputs, that is, the probability that a positive output is a TP, as opposed to an FP.

prediction. In the context of SL, the task of determining what numerical output is appropriate for a given input. Prediction from numerical inputs is called regression.

probability of detection (P_d). See "recall."

procedural AI. Also called symbolic AI, a form of AI that manipulates variables and concepts that are meaningful to humans using logic and data structures implemented in traditional software.

prompt. In GenAI, a text string in natural language used to initiate the generation of model output. It can take the form of the beginning of a text or a question or instructions for what to produce. “Prompt engineering” is the study of how the quality of model outputs depends on how the prompt is phrased.

recall. In classification, the fraction of ground truth positives that are labeled as positive by the classifier. Other names for this metric are “sensitivity” and “probability of detection.”

receiver-operating characteristic (ROC) curve. Generally referred to by the acronym ROC (pronounced “rock”), a plot showing the trade-off of FP rate versus FN rate as the detection threshold of a sensor is varied. In ML, it is used to show how training with different objective functions that emphasize different parts of the confusion matrix will result in a similar trade-off.

regression. In ML, a generic term for any SL model that predicts numerical outputs based on numerical inputs.

regression testing. In T&E, the retesting of a previously tested system or component after a change in design or in mission environment using inputs for which results of previous tests are known. This use of the term “regression” is unrelated to the ML sense of a model that predicts numerical outputs from numerical inputs.

reinforcement learning (RL). An ML technique in which a model is taught to perform a complex task by formulating a parametric policy that describes what action to take in each possible state of the task. The model is trained through repeated trials of the task with feedback from a reward function.

representativeness. In ML, the extent to which the TVT data have the same distribution and correlations of features and labels as the intended mission environment.

residual(s). In regression, the errors in individual predictions on the TVT data, that is, the differences between the predicted values and the corresponding ground truth values.

reward function. In RL, the reward function scores how well a proposed policy has performed its intended task in one iteration of the learning algorithm.

reward hacking. In RL, reward hacking is when the model exploits an imperfection in how the state space and reward function capture real-world behaviors to learn a policy that gets high rewards in training but is not effective in actual operations.

robustness. The ability of a model or system to avoid unacceptably poor performance regardless of inputs or mission context. ML models that are not explicitly trained to be robust tend to exhibit poor worst-case behavior.

sensitivity. See “recall.”

strategy. In the context of RL, a policy or strategy is a parametric representation of how the RL model will respond to being in a particular state and receiving a particular input.

supervised learning (SL). A form of ML that trains a model to classify or predict based on a set of labeled T&V data instances for which ground truth is known.

synthetic data. TVT data that were not collected from the real world. They can be generated by simulation models or by combining or perturbing real-world data.

test dataset. In the TVT dataset, the test dataset contains the instances that will be used to assess the quality of the trained model at the end of the training algorithm. These instances are not used during the training process.

testing data. See “training, validation, and test (TVT) data.”

training data. See “training, validation, and test (TVT) data.”

training, validation, and test (TVT) data. In SL and UL, models are developed using large numbers of cleaned, normalized, and standardized data instances in a standard format and representation. A subset of the data is set aside as the test set, which is not used in training. The remaining data are separated into a training set and a validation set for each iteration of the learning algorithm.

true negative (TN). For a classifier, a TN occurs when the classifier says that a particular label does not apply to the input instance, and this is correct.

true positive (TP). For a classifier, a TP occurs when the classifier says that a particular label applies to the input instance, and this is correct.

unsupervised learning (UL). A form of ML that trains a model using unlabeled data instances for which no ground truth grouping or status is known. Important uses of UL include clustering and anomaly detection.

unwanted bias. Bias occurs when the outputs of an ML model consistently give more favorable outcomes to some identifiable groups than to others. This bias typically arises from correlations in the training data. Unwanted bias is any model bias that discriminates in ways that are not intended and not desired by the stakeholders of the model’s application. Avoiding unwanted bias is one of the DoD AI Ethical Principles.

validation. The process of determining the degree to which a model, simulation, or distributed simulation, and associated data are an accurate representation of the real world from the perspective of the specific intended use. Validation across the M&S life cycle entails application of relevant referent data to refine M&S accuracy. (DoDI 5000.61) For TVT data, validation means establishing that the data support the training of a mission-appropriate model for the intended use and environment.

validation data. See “training, validation, and test (TVT) data.”

verification. The process of determining that a model, simulation, or distributed simulation, and associated data accurately represent the developer’s conceptual description and specifications (DoDI 5000.61). For TVT data, verification means establishing that the data are of sufficient quality to train a credible ML model.

VV&A. Verification, validation, and accreditation. See those entries for additional information.

Acronyms

ACAT	Acquisition Category
AI	artificial intelligence
AIEC	artificial intelligence-enabled capability
AIES	artificial intelligence-enabled system(s)
AoA	analysis of alternatives
APA	additional performance attribute
ATP	authority to proceed
AWS	autonomous weapon system(s)
CDAO	Chief Digital and Artificial Intelligence Office
COI	critical operational issue
CONEMP	concept of employment
DBS	defense business system(s)
DCAPE	Director of Cost Assessment and Program Evaluation
DepSecDef	Deputy Secretary of Defense
DFARS	Defense Federal Acquisition Regulation Supplement
DoD	Department of Defense
DoDD	DoD directive
DoDI	DoD instruction
DOE	design of experiments
DT	developmental test/testing
DT&E	developmental test and evaluation
DTE&A	Developmental Test, Evaluation, and Assessments
EMD	engineering and manufacturing development
FAR	Federal Acquisition Regulation
F-M	Fowlkes-Mallows
FN	false negative
FP	false positive
FRP	full-rate production

GAI	generative AI
GenAI	generative AI
HMI	human-machine interaction
HMT	human-machine team/teaming
HSI	human-systems integration
HTML	Hypertext Markup Language
ICE	independent cost estimate
IDSK	Integrated Decision Support Key
IP	intellectual property
IT	information technology
ITRA	Independent Technical Risk Assessment
JATIC	Joint AI Test Infrastructure Capability
JCIDS	Joint Capabilities Integration and Development System
KPP	key performance parameter
KSA	key system attribute
LIME	local interpretable model-agnostic explanations
LLM	large language model
M&S	modeling and simulation
MAE	mean absolute error
MCA	major capability acquisition
MDAP	major defense acquisition program
MDD	Materiel Development Decision
ML	machine learning
MLES	machine learning-enabled system(s)
MSE	mean squared error
MTA	middle tier of acquisition
NIST	National Institute of Standards and Technology
OCR	optical character recognition
OMB	Office of Management and Budget

OODA	observe-orient-decide-act
OSD	Office of the Secretary of Defense
OT	operational test/testing
OT&E	operational test and evaluation
OTRR	Operational Test Readiness Review
OUSD(R&E)	Office of the Under Secretary of Defense for Research and Engineering
OWL	Web Ontology Language
PCA	principal components analysis
P_d	probability of detection
PM	program manager
R^2	coefficient of determination
RAI	responsible AI
RFP	request for proposals
RL	reinforcement learning
RMF	Risk Management Framework
ROC	receiver-operating characteristic
RTA	runtime assurance
S&T	science and technology
SA	situational awareness
SI	systems integration
SL	supervised (and semi-supervised) learning
SME	subject matter expert
SWaP-C	space, weight, power, and cooling
T&E	test and evaluation
T&V	training and validation
TEMP	test and evaluation master plan
TES	test and evaluation strategy
TMRR	technology maturation and risk reduction
TN	true negative

TP	true positive
TRA	Technology Readiness Assessment
TVT	training, validation, and test
UL	unsupervised learning
U.S.C.	United States Code
USD(A&S)	Under Secretary of Defense for Acquisition and Sustainment
USD(P)	Under Secretary of Defense for Policy
USD(R&E)	Under Secretary of Defense for Research and Engineering
V&V	verification and validation
VCJCS	Vice Chairman of the Joint Chiefs of Staff
VV&A	verification, validation, and accreditation
XAI	explainable AI

References

AcqNotes, The Defense Acquisition Encyclopedia, s.v. “Data Rights,” updated August 22, 2023. https://acqnotes.com/acqnote/careerfields/data-rights

Artificial Intelligence Acquisition Guidebook. Department of the Air Force/Massachusetts Institute of Technology. Cambridge, MA: DAF/MIT Artificial Intelligence Accelerator, February 14, 2022. https://aia.mit.edu/wp-content/uploads/2022/02/AI-Acquisition-Guidebook_CAO-14-Feb-2022.pdf

Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. National Institute of Standards and Technology, January 2023. https://doi.org/10.6028/NIST.AI.100-1

Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. National Institute of Standards and Technology, July 2024. https://doi.org/10.6028/NIST.AI.600-1

Badillo, Solveig, Balazs Banfai, Fabian Birzele, et al. “An Introduction to Machine Learning.” *Clinical Pharmacology and Therapeutics* 107(4): 871–885, 2020. https://doi.org/10.1002/cpt.1796

Balhana, Carlos, Ivy Chen, Ronald W. Ferguson, et al. “Systems Engineering Processes to Test AI Right (SEPTAR) Release 1.” MITRE Technical Report, August 2023. https://apps.dtic.mil/sti/trecms/pdf/AD1211716.pdf

Banh, Leonardo, and Gero Strobel. “Generative Artificial Intelligence.” *Electronic Markets* 33(63), December 6, 2023. https://doi.org/10.1007/s12525-023-00680-1

Bunel, Rudy, Ilker Turkaslan, Philip H.S. Torr, Pushmeet Kohli, and M. Pawan Kumer. “A Unified View of Piecewise Linear Neural Network Verification.” arXiv, May 22, 2018. https://arxiv.org/pdf/1711.00455

Cai, Bryan, Fabio Pellegrini, Menglan Pang, et al. “Bootstrapping the Cross-Validation Estimate.” Cornell University arXiv, July 1, 2023. https://arxiv.org/abs/2307.00260

Celebi, M. Emre, and Kemal Aydin, editors. *Unsupervised Learning Algorithms.* Springer International Publishing Switzerland, 2016. https://doi.org/10.1007/978-3-319-24211-8

Chairman of the Joint Chiefs of Staff Instruction 3030.01A, “Implementing Joint Force Development and Design,” October 3, 2022.

Chandrasekaran, Jaganmohan, Tyler Cody, Nicola McCarthy, Erin Lanus, and Laura Freeman. “Test and Evaluation Best Practices for Machine Learning-Enabled Systems.” Cornell University arXiv, October 10, 2023. https://doi.org/10.48550/arXiv.2310.06800

Creswell, Antonia, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. “Generative Adversarial Networks: An Overview.” *IEEE Signal Processing Magazine* 35(1): 53–65, 2018. https://doi.org/10.1109/MSP.2017.2765202

Dalpe, Allisa J., May-Win L. Thein, and Martin Renken. “Perform: A Metric for Evaluating Autonomous System Performance in Marine Testbed Environments Using Interval Type-2 Fuzzy Logic.” *Applied Sciences* 11(24): 11940, 2021. https://doi.org/10.3390/app112411940

Damacharla, Praveen, Ahmad Y. Javaid, Jennie J. Gallimore, and Vijay K. Devabhaktuni. “Common Metrics to Benchmark Human-Machine Teams (HMT): A Review.” *IEEE Access*, 2018. https://doi.org/10.1109/ACCESS.2018.2853560

DataRobot. *Data Preparation*, 2024. https://www.datarobot.com/wiki/data-preparation/#:~:text=What%20is%20Data%20Preparation%20for,uncover%20insights%20or%20make%20predictions.&text=Improperly%20formatted%20%2F%20structured%20data

DeCostanza, Arwen H., Amar R. Marathe, Addison Bohannon, et al. “Enhancing Human-Agent Teaming with Individualized, Adaptive Technologies: A Discussion of Critical Scientific Questions.” Report ARL-TR-8359. U.S. Army Research Laboratory, May 2018. doi: 10.13140/RG.2.2.12666.39364

Defense Science Board Summer Study on Autonomy. Washington, D.C.: Office of the Under Secretary of Defense for Defense for Acquisition, Technology, and Logistics, June 2016. https://apps.dtic.mil/sti/pdfs/AD1017790.pdf

Department of Defense Cyber Developmental Test and Evaluation Guidebook, Version 3.0, currently under development to be published by OUSD(R&E)/DTE&A.

Department of Defense Cybersecurity Test and Evaluation Guidebook, Version 2.0, Change 1. Washington, DC: Department of Defense, February 10, 2020. https://www.dau.edu/sites/default/files/2023-09/Cybersecurity-Test-and-Evaluation-Guidebook-Version2-change-1.pdf

Department of Defense Data, Analytics, and Artificial Intelligence Adoption Strategy: Accelerating Decision Advantage. Washington, DC: Department of Defense, 2023. https://media.defense.gov/2023/Nov/02/2003333300/-1/-1/1/DOD_DATA_ANALYTICS_AI_ADOPTION_STRATEGY.PDF

Department of Defense Interoperability Process Guide, Version 3.0. Joint Interoperability Test Command, July 2023. https://jitc.fhu.disa.mil/isg/downloads/IPG_Version_3_Final_signed_20230703.pdf

Department of Defense Responsible Artificial Intelligence Strategy and Implementation Pathway. Washington, DC: Department of Defense, June 2022. https://media.defense.gov/2022/Jun/22/2003022604/-1/-1/0/Department-of-Defense-Responsible-Artificial-Intelligence-Strategy-and-Implementation-Pathway.PDF

Deputy Secretary of Defense Memorandum, “Implementing Responsible Artificial Intelligence in the Department of Defense,” May 26, 2021.

https://media.defense.gov/2021/May/27/2002730593/-1/-1/0/IMPLEMENTING-RESPONSIBLE-ARTIFICIAL-INTELLIGENCE-IN-THE-DEPARTMENT-OF-DEFENSE.PDF

Dhavale, Sunita Vikrant. *Advanced Image-Based Spam Detection and Filtering Techniques.* Hershey, PA: IGI Global Scientific Publishing, 2017. https://doi.org/10.4018/978-1-68318-013-5

Director of the Office of Management and Budget Memorandum, "Advancing Governance, Innovation, and Risk Management for Agency Use of Artificial Intelligence," M-24-10, March 28, 2024.

DoD Directive 3000.09, "Autonomy in Weapon Systems," January 25, 2023.

DoD Directive 5137.02, "Under Secretary of Defense for Research and Engineering (USD(R&E))," July 15, 2020.

DoD Instruction 5000.02, "Operation of the Adaptive Acquisition Framework," January 23, 2020, as amended

DoD Instruction 5000.61, "DoD Modeling and Simulation Verification, Validation, and Accreditation," September 17, 2024.

DoD Instruction 5000.74, "Defense Acquisition of Services," January 10, 2020, as amended.

DoD Instruction 5000.75, "Business Systems Requirements and Acquisition," February 2, 2017, as amended.

DoD Instruction 5000.80, "Operation of the Middle Tier of Acquisition," December 30, 2019, as amended.

DoD Instruction 5000.81, "Urgent Capability Acquisition," December 31, 2019.

DoD Instruction 5000.84, "Analysis of Alternatives," August 4, 2020.

DoD Instruction 5000.85, "Major Capability Acquisition," August 6, 2020, as amended.

DoD Instruction 5000.87, "Operation of the Software Acquisition Pathway," October 2, 2020.

DoD Instruction 5000.DT, "Test and Evaluation," TBD.

DoD Instruction 8320.02, "Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense," August 5, 2013, as amended.

DoD Instruction 8320.07, "Implementing the Sharing of Data, Information, and Information Technology (IT) Services in the Department of Defense," August 3, 2015, as amended.

Gandon, Fabien, Reto Krummenacher, Sung-Kook Han, and Ioan Toma. *The Resource Description Framework and Its Schema*. Handbook of Semantic Web Technologies, 2011.

Gao, Leo, Stella Biderman, Sid Black, et al. "The Pile: An 800GB Dataset of Diverse Text for Language Modeling." arXiv:2101.00027, December 31, 2020. https://arxiv.org/pdf/2101.00027v1.pdf

Gehr, Timon, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. "AI2: Safety and Robustness Certification of Neural Networks with Abstract

Interpretation." *2018 IEEE Symposium on Security and Privacy*, San Francisco, CA: 3–18, 2018.

Geuvers, Herman. "Proof Assistants: History, Ideas and Future." *Sādhanā* 34(1): 3–25, 2009. https://doi.org/10.1007/s12046-009-0001-5

Goddard, Kate, Abdul Roudsari, and Jeremy C. Wyatt. "Automation Bias: A Systematic Review of Frequency, Effect Mediators, and Mitigators." *Journal of the American Medical Infomatics Association* 19(1): 121–127, 2012. https://doi.org/10.1136/amiajnl-2011-000089

Government Accountability Office Report GAO-23-105850, "Artificial Intelligence: DOD Needs Department-Wide Guidance to Inform Acquisitions," June 2023. https://www.gao.gov/assets/gao-23-105850.pdf

Grandini, Margherita, Enrico Bagli, and Giorgio Visani. "Metrics for Multi-Class Classification: An Overview." ArXiv, August 13, 2020. https://doi.org/10.48550/arXiv.2008.05756

Greiffenhagen, Christian. "Checking Correctness in Mathematical Peer Review." *Social Studies of Science* 54(2): 184–209, 2024.

Gross, Kerianne H., Matthew A. Clark, Jonathan A. Hoffman, Eric D. Swenson, and Aaron W. Fifarek. "Run-Time Assurance and Formal Methods Analysis Nonlinear System Applied to Nonlinear System Control." *Journal of Aerospace Information Systems* 14(4): 232–246, 2017. https://doi.org/10.2514/1.I010471

Guerraoui, Rachid, Nirupam Gupta, and Rafael Pinot. *Robust Machine Learning: Distributed Methods for Safe AI*. ISBN 978-981-97-0688-4. Springer Singapore, 2024. https://doi.org/10.1007/978-981-97-0688-4

Hawkins, Richard, Colin Paterson, Chiara Picardi, Yan Jia, Radu Calinescu, and Ibrahim Habli. Cornell University arXiv. "Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS)," February 2, 2021. https://doi.org/10.48550/arXiv.2102.01564

Hendrycks, Dan, and Thomas Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations." Cornell University arXiv, March 28, 2019. https://arxiv.org/pdf/1903.12261

Hoare, C. A. R. "An Axiomatic Basis for Computer Programming." *Communications of the ACM* 12(10): 576–580, 1969. https://doi.org/10.1145/363235.363259

Holzmann, Gerard J. *The Spin Model Checker: Primer and Reference Manual*, Addison-Wesley, 2003.

Hong, Yili, Jiayi Lian, Li Xu, et al. "Statistical Perspectives on Reliability of Artificial Intelligence Systems." *Quality Engineering* 35(1): 56–78, 2022. https://doi.org/10.1080/08982112.2022.2089854

Human-AI Teaming: State-of-the-Art and Research Needs. The National Academies of Sciences, Engineering, and Medicine. Washington DC: The National Academies Press, 2022. https://doi.org/10.17226/26355

Human Systems Integration Test and Evaluation of Artificial Intelligence-Enabled Capabilities. Chief Digital and Artificial Intelligence Office (CDAO), April 2024. https://cdao.pages.jatic.net/public/guidance/human-systems-integration/

Hutchinson, Ben, Negar Rostamzadeh, Christina Greer, Katherine Heller, and Vinodkumar Prabhakaran. “Evaluation Gaps in Machine Learning Practice.” In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 1859–1876, June 20, 2022. https://doi.org/10.1145/3531146.3533233

Igual, Laura, and Santi Seguí. *Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications*. 2nd ed. Springer Nature Switzerland, 2024. https://doi.org/10.1007/978-3-031-48956-3

Integrated Master Plan and Integrated Master Schedule Preparation and Use Guide. Washington, DC: Office of the Under Secretary of Defense for Research and Engineering, May 2023. https://ac.cto.mil/wp-content/uploads/2023/05/IMP-IMS-Guide-2023.pdf

Jussupow, Ekaterina, Izak Benbasat, and Armin Heinzl. “Why are We Averse Towards Algorithms? A Comprehensive Literature Review on Algorithm Aversion.” *Research Papers*, 2020. https://aisel.aisnet.org/ecis2020_rp/168

Koopman, Philip, and Michael Wagner. “Toward a Framework for Highly Automated Vehicle Safety Validation.” SAE Technical Paper 2018-01-1071, 2018. https://doi.org/10.4271/2018-01-1071

Kuzio de Naray, Rachel, George “Lee” Kennedy, Ryan Wagner, and Steven Wartik. “Cybersecurity and DoD System Development: A Survey of DoD Adoption of Best DevSecOps Practice.” IDA Document P-22749. Alexandria, VA: Institute for Defense Analyses, September 2021.

Lanus, Erin, Laura J. Freeman, D. Richard Kuhn, and Raghu N. Kacker. “Combinatorial Testing Metrics for Machine Learning.” *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Porto de Galinhas, Brazil, pp. 81–84, April 12–16, 2021. https://doi.org/10.1109/ICSTW52544.2021.00025

Li, Jianlin, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang. “Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification.” *26th Static Analysis Symposium*, Porto, Portugal, October 8–11, 2019.

Lyons, Joseph B., and Kevin T. Wynne. “Human-Machine Teaming: Evaluating Dimensions Using Narratives.” *Human-Intelligent Systems Integration* 3(2): 129–137, 2021. https://doi.org/10.1007/s42454-020-00019-7

Ma, Lanssie Mingyue, Martijn Ijtsma, Karen M. Feigh, and Amy R. Pritchett. “Metrics for Human-Robot Team Design: A Teamwork Perspective on Evaluation of Human-Robot Teams.” *ACM Transactions on Human-Robot Interaction* 11(3), Article 30: 1–36, 2022.

https://doi.org/10.1145/3522581

Manual for the Operation of the Joint Capabilities Integration and Development System. Joint Staff J-8 Force Structure, Resources, and Assessment Directorate, October 30, 2021.

Mao, Anqi, Mehryar Mohri, and Yutao Zhong. "Cross-Entropy Loss Functions: Theoretical Analysis and Applications." Proceedings of *The 40th International Conference on Machine Learning*, PMLR 202, 2023.

McAdams, Mindy. "Symbolic AI: Good Old-Fashioned AI." AI in Media and Society blog entry, June, 3, 2021. https://www.macloo.com/ai/2021/06/03/symbolic-ai-good-old-fashioned-ai/

McDonald, Gary C. "Ridge Regression." *Wiley Interdisciplinary Reviews (WIREs): Computational Statistics* 1(1): 93–100, 2009. https://doi.org/10.1002/wics.14

Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. "A Survey on Bias and Fairness in Machine Learning." *ACM Computing Surveys*, Volume 54, Issue 6, Article No. 115, pp. 1–35, 2021. https://doi.org/10.1145/3457607

Metz, Cade. "Google is 2 Billion Lines of Code—And It's All in One Place." *WIRED*, September 16, 2015. https://www.wired.com/2015/09/google-2-billion-lines-codeand-one-place/

Military Standard MIL-STD-882E, "System Safety," September 27, 2023.

Mullins, Galen E., Paul G. Stankiewicz, R. Chad Hawthorne, et al. "Delivering Test and Evaluation Tools for Autonomous Unmanned Vehicles to the Fleet." *Johns Hopkins APL Technical Digest*, Volume 33, Number 4, 2017.

Murphy, Christian, Gail E. Kaiser, and Lifeng Hu. "Properties of Machine Learning Applications for Use in Metamorphic Testing." Department of Computer Science, Columbia University, 2011. https://doi.org/10.7916/D8XK8PFD

Nagubandi, Govind. "Field Guide to Address Bias in Datasets." Penn Law Policy Lab on AI and Bias, April 2021. https://www.law.upenn.edu/live/files/11569-field-guide-to-address-bias-in-datasets

Nikolenko, Sergey I. *Synthetic Data for Deep Learning*. Springer Nature Switzerland, 2021. https://doi.org/10.1007/978-3-030-75178-4

Operational Test and Evaluation of Artificial Intelligence-Enabled Capabilities. Chief Digital and Artificial Intelligence Office (CDAO), April 2024. https://cdao.pages.jatic.net/public/guidance/operational/

O'Regan, Gerard. *Concise Guide to Formal Methods: Theory, Fundamentals and Industry Applications*. Springer Nature Switzerland, 2017. https://link.springer.com/content/pdf/10.1007/978-3-319-64021-1.pdf

Political Declaration on Responsible Military Use of Artificial Intelligence and Autonomy. Washington DC: U.S. Department of State, November 9, 2023.

Pulina, Luca, and Armando Tacchella. “An Abstraction-Refinement Approach to Verification of Artificial Neural Networks.” *22nd Computer Aided Verification International Conference*, Edinburgh, UK, 2010.

Pulina, Luca, and Armando Tacchella. “Challenging SMT Solvers to Verify Neural Networks.” *AI Communications* 25(2): 117–135, 2012.

Pullum, Laura L. “Review of Metrics to Measure the Stability, Robustness and Resilience of Reinforcement Learning.” Cornell University arXiv, March 22, 2022. https://doi.org/10.48550/arXiv.2203.12048

Raistrick, Chris, Paul Francis, John Wright, Colin Carter, and Ian Wilkie. *Model Driven Architecture with Executable UML*, Cambridge University Press, 2004.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. “Model-Agnostic Interpretability of Machine Learning.” *2016 ICML Workshop on* Human *Interpretability in Machine Learning*. New York, NY, 2016. https://arxiv.org/abs/1606.05386v1

Ryseff, James, Brandon F. De Bruhl, and Sydne J. Newberry. “The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed: Avoiding the Anti-Patterns of AI.” RAND Corporation, August 13, 2024. https://www.rand.org/pubs/research_reports/RRA2680-1.html

Savage, L. J. “The Theory of Statistical Decision.” *Journal of the American Statistical Association* 46(253): 55–67, 1951. https://doi.org/10.2307/2280094

Scheibler, Karsten, Leonore Winterer, Ralf Wimmer, and Bernd Becker. “Towards Verification of Artificial Neural Networks.” *MBMV*: 30–40, 2015.

Schieferdecker, Ina, and Andreas Hoffmann. “Model-Based Testing.” *Encyclopedia of Software Engineering*, pp. 1–15, 2011.

Schierman, John D., Michael D. DeVore, Mathan D. Richards, and Matthew A. Clark. “Runtime Assurance for Autonomous Aerospace Systems.” *Journal of Guidance, Control, and Dynamics* 43(12): 2205–2217, 2020. https://doi.org/10.2514/1.G004862

“Secretary of Defense Memorandum, “Artificial Intelligence Ethical Principles for the Department of Defense,” February 21, 2020”

Seto, Danbing, Bruce H. Krogh, Lui Sha, and Alongkrit Chutinan. “Dynamic Control System Upgrade Using the Simplex Architecture.” *IEEE Control Systems Magazine* 18(4): 72–80, 1998. https://doi.org/10.1109/37.710880

Soklaski, Ryan, Justin Goodwin, Olivia Brown, Michael Yee, and Jason Matterer. “Tools and Practices for Responsible AI Engineering.” Cornell University arXiv, January 14, 2022. https://arxiv.org/abs/2201.05647

Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz. “Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation.” *AI 2006: Advances in Artificial Intelligence*, 1015–1021, 2006.

Sparrow, David A., David M. Tate, John C. Biddle, Nicholas J. Kaminski, and Poornima Madhavan. “Assessing the Quality of Decision-making by Autonomous Systems.” IDA Document P-9116. Alexandria, VA: Institute for Defense Analyses, 2018.

Sun, Shiliang, Zehui Cao, Han Zhu, and Jing Zhao. “A Survey of Optimization Methods from a Machine Learning Perspective.” *IEEE Transactions on Cybernetics* 50(8): 3668–3681, 2020. https://doi.org/10.1109/TCYB.2019.2950779

Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. The MIT Press, 2018.

Systems Integration Test and Evaluation of Artificial Intelligence-Enabled Capabilities. Chief Digital and Artificial Intelligence Office (CDAO), April 2024. https://cdao.pages.jatic.net/public/guidance/systems-integration/

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, et al. “Intriguing Properties of Neural Networks.” *2nd International Conference on Learning Representations (ICLR) 2014*, Banff, Alberta, Canada, 2014.

Tate, David M. “Trust, Trustworthiness, and Assurance of AI and Autonomy.” IDA Document D-22631. Alexandria, VA: Institute for Defense Analyses, April 2021.

Test and Evaluation as a Continuum: The Integrated Decision Support Key (IDSK) for Test Planning. Defense Acquisition University. Webinar, November 17, 2023. https://media.dau.edu/media/t/1_6lv01c7e

Test and Evaluation Strategy Frameworks. Washington, DC: Chief Digital and Artificial Intelligence Office (CDAO). https://cdao.pages.jatic.net/public/guidance/

Tibshirani, Robert. “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1): 267–288, 1996.

Trusilo, Daniel. “Autonomous AI Systems in Conflict: Emergent Behavior and Its Impact on Predictability and Reliability.” *Journal of Military Ethics* 22(1): 2–17, 2023. https://doi.org/10.1080/15027570.2023.2213985

Tulbure, Andrei-Alexandru, Adrian-Alexandru Tulbure, and Eva-Henrietta Dulf. “A Review on Modern Defect Detection Models using DCNNs – Deep Convolutional Neural Networks.” *Journal of Advanced Research* 35: 33–48, 2022. https://doi.org/10.1016/j.jare.2021.03.015

UL 4600, “Standard for Safety for the Evaluation of Autonomous Products.” 3rd ed. UL Standards and Engagement, March 17, 2023.

United States Code, Title 10

Urban, Caterina, and Antoine Miné. “A Review of Formal Methods applied to Machine Learning.” Cornell University arXiv, April 21, 2021. https://doi.org/10.48550/arXiv.2104.02466

Vice Chairman of the Joint Chiefs of Staff Memorandum, “Conventional Arms Transfer Policy, National Security Presidential Memorandum Task 2.7, Build Exportability,” April 15, 2019. https://www.dau.edu/sites/default/files/Migrated/CopDocuments/JROCM%20025-19%20CAT%20Policy%20NSPM%20Task%202.7%20Build%20Exportability.pdf

Wang, Shiqi, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. “Efficient Formal Safety Analysis of Neural Networks.” *Advances in Neural Information Processing Systems* 31, 2018.

Webb, Geoffrey I., Loong Kuan Lee, Bart Goethals, and François Petitjean. “Analyzing Concept Drift and Shift from Sample Data.” *Data Mining and* Knowledge *Discovery* 32: 1179–1199, 2018. https://doi.org/10.1007/s10618-018-0554-1

Wilkins, Jay, David A. Sparrow, Caitlan A. Fealing, Brian D. Vickers, Kristina A. Ferguson, and Heather Wojton. “A Team-Centric Metric Framework for Testing and Evaluation of Human-Machine Teams.” *Systems Engineering* 27(3): 466–484, 2024. https://doi.org/10.1002/sys.21730

Wojton, H. et al. “Characterizing Human-Machine Teaming Metrics for Test and Evaluation.” Document NS D-14349. Alexandria, VA: Institute for Defense Analyses, 2020.

Xiang, Weiming, Hoang-Dung Tran, and Taylor T. Johnson. “Output Reachable Set Estimation and Verification for Multi-Layer Neural Networks.” *IEEE Transactions on Neural Networks and Learning Systems* 29(11): 5777–5783, 2018.

Xu, Dongkuan, and Yingjie Tian. “A Comprehensive Survey of Clustering Algorithms.” *Annals of Data Science* 2: 165–193, 2015. https://doi.org/10.1007/s40745-015-0040-1

Websites

CDAO JATIC Documentation, Guidance. https://cdao.pages.jatic.net/public/guidance/

CDAO RAI Toolkit. https://rai.tradewindai.com/

The Data Cards Playbook. https://sites.research.google/datacardsplaybook/

DoD Issuances. https://www.esd.whs.mil/DD/DoD-Issuances/

Model Cards. https://www.kaggle.com/code/var0101/model-cards

VV&A Recommended Practices Guide. https://www.cto.mil/sea/vva_rpg/

W3C OWL Web Ontology Language Overview. https://www.w3.org/TR/owl-features/

Developmental Test and Evaluation of Artificial Intelligence-Enabled Systems

Office of the Director, Developmental Test, Evaluation, and Assessments
Office of the Under Secretary of Defense for Research and Engineering
3030 Defense Pentagon
Washington, DC 20301
osd.r-e.comm@mail.mil
https://www.cto.mil/dtea/