
DEVCOM DAC-TR-2023-084
December 2023

# Ensured Reliability for Artificial Intelligence Systems

by Nathan Herbert, Barry Hum, Patiana Theragene, and
Alexander Marchioni

## DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so specified by other official documentation.

## WARNING

Information and data contained in this document are based on the input available at the time of preparation.

## TRADE NAMES

The use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software. The report may not be cited for purposes of advertisement.

DEVCOM DAC-TR-2023-084
December 2023

# Ensured Reliability for Artificial Intelligence Systems

**by Nathan Herbert, Barry Hum, Patiana Theragene, and Alexander Marchioni**
***DEVCOM Analysis Center***

# REPORT DOCUMENTATION PAGE

| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED | |
|---|---|---|---|
| | | START DATE | END DATE |
| December 2023 | Technical Report | 10/1/2022 | 9/30/2023 |

**4. TITLE AND SUBTITLE**
Ensured Reliability for Artificial Intelligent Systems

| 5a. CONTRACT NUMBER | 5b. GRANT NUMBER | 5c. PROGRAM ELEMENT NUMBER |
|---|---|---|
| **5d. PROJECT NUMBER** | **5e. TASK NUMBER** | **5f. WORK UNIT NUMBER** |

**6. AUTHOR(S)**
Nathan Herbert, Barry Hum, Patiana Theragene, and Alexander Marchioni

| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) | 8. PERFORMING ORGANIZATION REPORT NUMBER |
|---|---|
| Director<br>DEVCOM Analysis Center<br>6896 Mauchly Street<br>Aberdeen Proving Ground, MD 21005 | DEVCOM DAC-TR-2023-084 |

| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |
|---|---|---|
| | | |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
Artificial intelligence (AI)/assistive automation applications are becoming more common in the Army's developmental technologies and systems. Ensuring these systems perform reliably is critical to the success of the Army's evolving mission. In addition to traditional hardware and software aspects, development will also need to account for human and environmental interactions within the systems. New areas regarding the data pipeline and AI algorithm training become instrumental in the reliability of AI systems. Previous reports have examined AI-specific design-for-reliability activities and identified potential failure modes in AI systems. This report expands on those efforts and discusses AI system elements that are necessary to mitigate reliability risks. These elements will form the basis for future reliability risk assessment methodology.

**15. SUBJECT TERMS**
reliability, artificial intelligence, AI, risk, autonomy, best practices

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES |
|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | C. THIS PAGE | | |
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UU | 34 |

| 19a. NAME OF RESPONSIBLE PERSON | 19b. PHONE NUMBER *(Include area code)* |
|---|---|
| Nathan Herbert | (410) 278-5169 |

**STANDARD FORM 298 (REV. 5/2020)**
*Prescribed by ANSI Std. Z39.18*

# Table of Contents

# List of Figures

# List of Tables

# 1. INTRODUCTION

Artificial intelligence (AI) applications and autonomy are becoming more common in the Army's developmental technologies and systems. Ensuring these systems perform reliably is critical to the success of the Army's evolving mission. In addition to traditional hardware and software aspects, development will also need to account for human and environmental interactions within the systems. New areas regarding the data pipeline and AI algorithm training become instrumental in the reliability of AI systems.

AI systems, as used in this report, are complex electromechanical systems that leverage AI and include human operators. As a result, AI systems are essentially comprised of hardware, software, operator, and a type of machine learning (ML) algorithm. The AI/ML algorithm is software as well, but it can take on other attributes or characteristics that are different than those of software. A system containing an AI/ML algorithm may have the ability to perform tasks commonly associated with human cognition, as learning, reasoning, and decision-making become critical parts of the AI system and its reliability.

Typically, system reliability is associated with performing a mission without failure. For AI systems, performing a mission without failure can involve other system attributes besides reliability. The definition of failure expands to include aspects of these other system attributes. There are elements that developers should focus on to mitigate failures and help ensure mission completion.

The rest of this report is organized as follows. Section 2 presents an overview of the relationship between AI system reliability and two other attributes: robustness and resilience. Section 3 provides a synopsis of AI failure modes presented in previous research. Section 4 discusses some of the key elements supporting reliability that developers should consider in AI system development. Finally, Section 5 contains conclusions and the next steps for future work in this area.

# 2. RELIABILITY, ROBUSTNESS, AND RESILIENCE

In systems engineering, reliability is defined as the probability that a system will function as expected under specified conditions for a given operational period. This definition fits the traditional hardware and software components of an AI system. For example, a hardware panel of an airplane is designed to be reliable in the conditions that an airplane sees—namely, altitudes between 0 and 35,000 ft. But that same panel may not function as expected if it is used outside those conditions, like on a spacecraft. And unlike an AI algorithm, that same panel is not intelligent and cannot adjust to maintain its level of reliability in those new environments.

While AI systems may be designed for certain uses and conditions, they need to be less rigid than traditional software and hardware systems. They need to be adaptable to changing environments and operating conditions, like how human intelligence is adaptable. For example, let us look at a human driving a vehicle. A U.S. resident learns, trains, and operates a vehicle based on the traffic designs instituted by U.S. road designs and laws. Now, put them in a car in Europe, where the driver sits on the opposite side of the vehicle and drives on the opposite side of the road, in comparison to the United States. While it may seem odd at first, a trained human can be expected to easily adapt to these new operating conditions and environments. For AI systems to be useful, their AI/ML algorithms need to contain attributes that support this adaptability.

One attribute for a useful AI system is AI/ML robustness. Robustness can be defined as the ability of a system to maintain regular and anticipated function, despite exceptional, unforeseen events, stressful conditions such as component failures, loss of service, or extreme conditions beyond the expected operating environment (Narayanan & Carrier, 2021). The International Organization for Standardization (ISO) defines AI robustness as the ability of an AI system to maintain its level of performance under any circumstances (ISO/IEC, 2022a). When unexpected data is encountered in operation, or the model is operating in less-than-ideal conditions, the robust AI tool continues to deliver accurate outputs. When an AI system encounters unexpected data or environmental conditions beyond what are expected, problems can arise if the system is not robust enough to handle them.

Another attribute is AI/ML algorithm resilience. Resilience is the speed and capability of the system to recover from major disruption (at a data level, model and pipeline level, and information level) to a sufficient level of function in accordance with the system's intended operation (Narayanan & Carrier, 2021). The ISO defines AI resilience as the ability of a system to maintain its level of performance under a variety of circumstances (ISO/IEC, 2022a).

AI systems in the military are expected to contribute to mission completion and mission success. Among other things, the mission success is dependent on traditional hardware, software, and operator reliability, as well as AI system robustness and resilience. DOD and industry are beginning to associate these principles together.

The “reliable” principle of the DOD’s Responsible AI Strategy states that 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 their entire lifecycles (DOD, 2022). As part of their research, Carnegie Mellon University (CMU) states that robust and secure AI systems are AI systems that reliably operate at expected levels of performance, even when faced with uncertainty and in the presence of danger or threat (2021). Finally, an industry leader explains that AI systems should operate reliably, safely, and consistently under normal circumstances and in unexpected conditions. AI can be called reliable if it deals with all the “what ifs” and adequately responds to the new situation without doing harm to users.

Considering the relationship between reliability, robustness, and resilience, the term “failure” can refer to more than traditional hardware and software anomalies. A failure can be referred to as an event when the AI, algorithm, or autonomous system does not perform as designed, including an incorrect prediction or statistically different prediction for a given instance or inability to guard against an adversary (Narayanan & Carrier, 2021).

# 3. AI FAILURE MODES

AI systems contain failure modes common in traditional, non-AI systems (e.g., hardware and software failures). Previous U.S. Army Combat Capabilities Development Command (DEVCOM) Analysis Center (DAC) research identified other failure modes that may be more impactful to AI systems than non-AI systems. This research highlighted three common themes in AI failures: data pipeline failures, human–machine failures, and intentional (adversarial) failures. Many of the failures associated with these failure modes impact the reliability, robustness, and resilience of the AI system.

## 3.1 Data Pipeline Failures

Many AI systems are based on models developed using training data that are the result of collecting, labeling, formatting, and other processes. This data pipeline is continued to be used in testing and in production for the system's useful life. Data pipelines are an important part of the AI system and are one of the main sources of failure modes. Table 1 presents data pipeline failures that were documented in previous DAC research (Blood et al., 2023).

**Table 1. Data pipeline failures**

| Failure mode | Description (with selected examples/scenarios) |
|---|---|
| 1. Performance bias | Unexpected/(undetected) discrepancies between data subsets.<br>Scenario: A targeting model has acceptable performance overall, but it performs much worse for some classes of targets than others because of differences in how the training data were formatted for the various classes. |
| 2. Out-of-distribution (distributional shifts) | Changes in the nature of the data.<br>Examples: Missing features, type/unit changes, out-of-range values, or sensor updates/calibration. |
| 2.1. Training-production skew | The training environment differs from the production environment.<br>Scenarios: 1) System designers make assumptions in the training data that do not apply to the operational environments. 2) The training environments do not cover the full operational space. |
| 2.1.1. Robustness failure (common corruption) | Common environmental noise/perturbations not included in training data.<br>Scenario: An image recognition system incorrectly classifies images that are subtly distorted (e.g., through noise/tilt/stretching). |
| 2.1.2. Target leakage | Unknown to the developers, the training data contain artifacts that improve AI performance, but those artifacts are not available in all operational environments.<br>Scenario: For a binary classifier, the positive cases contain a feature that the negative cases do not. |
| 2.2. Data drift | The operational environment changes over time.<br>Scenario: The false-negative rate in a tax evasion detection model increases during a period of higher inflation. |
| 2.3. Automated naivete | AI acts on inputs without scrutinizing their validity.<br>Scenario: An AI system keeps using sensor data after the sensor has failed. |
| 3. Cascading model failure | Modifications to a model have a negative effect on other models that use its output as their input.<br>Scenario: An AI model forecasting customer traffic feeds directly into a staffing AI model. After an improvement to the forecasting model, the staffing model experiences a drop in performance because it was not retrained using the outputs of the updated forecasting model. |
| 4. Feedback loop (side effects) | Model interaction with its environment affects incoming data in such a way that model performance is degraded or becomes biased.<br>Scenario: Spammers adjust emails to evade automated spam filters. |
| 5. Reward hacking | A discrepancy between model target and true desired target leads the model to behave in undesired ways.<br>Scenario: Modelers use a surrogated measure of performance that does not always correspond to desired performance for certain edge cases, and the model performs unexpectedly for those cases. |

## 3.2 Human–System Failures

The use of AI or assistive automation (AA) in human-operated systems creates a more collaborative dynamic between the operator and the rest of the system. This new relationship can lead to new failure modes and new perspectives on existing failure modes. In particular, the operator's cognitive load and trust levels in the AI system can

lead to faulty decisions and poor operational outcomes. Table 2 presents human–system failures that were documented in previous DAC research (Blood et al., 2023).

**Table 2. Human–system failures**

| Failure mode | Description (with selected examples/scenarios) |
|---|---|
| 6. Cognitive load | Failures resulting from shifts in the cognitive load placed on the operator. |
| 6.1. Up-tasking | The AI automates rote tasks, leaving operator mostly with tasks that require high-cognitive loads, resulting in cognitive fatigue.<br>Scenario: A single operator oversees a fleet of autonomous vehicles, intervening as needed. The need for intervention comes in a steady stream. Because most interventions are high load, the operator becomes burned out and begins making uncharacteristic mistakes. |
| 6.2. Atrophy | The operator's role is primarily an "AI overseer" for so long that skills degrade from disuse. The operator underperforms when intervention is needed. |
| 6.3. Passivity | The operator, whose role is primarily an "AI overseer," frequently becomes distracted by other things and underperforms when intervention is needed. |
| 7. Misunderstanding | Failures that occur when human operators do not correctly assess the state of the AI. |
| 7.1. Black-boxing | Prematurely "walling off" human operators from automated tasks.<br>Scenario: Based on an AI system's performance during testing, the operator is removed from the loop of one or more processes. The operator's actions within other processes create an unsafe situation due to misjudging the system state. |
| 7.2. Sneaky automation | Automating a task without incorporating the details of the automation in the documentation/training. |
| 7.3. Target mismatch | The operator is interested in a (sometimes subtly) different target than what the model was trained to predict or optimize, but this is not clear to the operator. |
| 8. Trust | Failures that occur when the operator's belief in the level of the AI system's performance is misplaced. |
| 8.1. Overconfidence | The operator develops an overconfidence in an AI system and does not respond appropriately when it behaves erroneously. |
| 8.2. Unrealistic expectations | The operator routinely ignores or overrides mostly correct output from the AI system because it is not always correct. |
| 8.3. Mistrust | The operator ignores or overrides correct AI decisions because the system fails to communicate the reason for its decisions. |

## 3.3 Intentional Failures

While the previous collections of failure modes are considered unintentional failures, there is a different collection of failures that are intentionally caused by adversaries. These intentional failures are the result of bad actors who compromise a system or manipulate it in undesirable ways. These failure modes are highly dependent on the system and its threat environment. A growing body of research into these failure modes underscores the importance of activities and collaborations focused on cybersecurity when designing in and testing for reliability. Through their research, Microsoft Corporation summarized their findings on intentionally motivated failures, as presented in Table 3 (Kumar & Snover, 2022).

**Table 3. Intentionally motivated failures**

| Failure mode | Description (with selected examples/scenarios) |
|---|---|
| 9. Perturbation attack | Attacker modifies the query to get an appropriate response. |
| 10. Poisoning attack | Attacker contaminates the training phase of ML systems to get intended result. |
| 11. Model inversion | Attacker recovers the secret features used in the mode through careful queries. |
| 12. Membership inference | Attacker can infer if given data record was part of the model's training data set or not. |
| 13. Model stealing | Attacker is able to recover the model by constructing careful queries, |
| 14. Reprogramming ML system | Repurpose the ML system to perform an activity it was not programmed for, |
| 15. Adversarial example in physical domain | Attacker brings adversarial examples into the physical domain to subvert ML system (e.g., 3D printing special eyewear to fool facial recognition system). |
| 16. Malicious ML provider recovering training data | Malicious ML provider can query the model used by customer and recover customer's training data. |
| 17. Attacking the ML supply chain | Attacker compromises the ML models as it is being downloaded for use. |
| 18. Backdoor ML | Malicious ML provides backdoor algorithm that does not work unless triggered. |
| 19. Exploit software dependencies | Attacker uses traditional software exploits like buffer overflow to confuse ML systems. |

# 4. ELEMENTS ENSURING AI SYSTEM RELIABILITY

Traditional non-AI Army systems are typically developed to address capability gaps. Recently, acquisition has become more agile to keep pace with technological advancements. For example, the Army's tactical network is constantly trying new commercial-off-the-shelf items to enhance network performance and reliability. With the emergence of AI, all commodity areas are exploring ways to incorporate it into their weapon systems or toolkits in support of weapon systems. When these AI systems are being developed, there are elements that should be considered to ensure reliable, robust, and resilient AI systems. The rest of this section highlights several of these key elements. This is not an exhaustive list of elements, but it does form the basis for early examination to assist in assessing reliability risks.

## 4.1 Use Cases

Unlike traditional systems that simply fill a capability gap, there are many reasons why more Army weapon systems are incorporating AI. Some AI systems are developed to improve performance and accuracy of military operations. Others are developed to perform missions where the Soldier can be removed from harm's way. Some are developed to help assess readiness, at both the tactical and force structure levels. All of these can be referred to as use cases for AI.

Much of the military AI and autonomy are being developed in the science and technology (S&T) community and experimental working groups. Table 4 summarizes many of the current military AI application categories.

**Table 4. Military use categories**

| Military use categories | Description (with selected examples/scenarios) |
|---|---|
| Autonomous vehicles | AI-driven autonomous vehicles, such as unmanned ground vehicles (e.g., a robotic combat vehicle) and unmanned aerial vehicles (e.g., Stingray–aerial refueling drone), can be used for surveillance, reconnaissance, refueling, and transportation in hazardous environments. |
| Autonomous weapons | AI-powered autonomous drones, tanks, launchers, and other weapons systems can be used for reconnaissance, surveillance, and even combat. These systems can make decisions without human intervention, which raises ethical and legal questions. |
| Counter-drone systems | AI can be used to detect and counter hostile drones, which have become a significant threat on the modern battlefield. |
| Contested logistics and supply chain management | AI can optimize the movement of troops and supplies, ensuring that resources are allocated efficiently and effectively, especially in large-scale military operations. |
| Target identification | AI can be used to analyze vast amounts of sensor data, such as imagery from satellites and drones, to identify and track potential targets, including enemy forces, vehicles, and infrastructure. |
| Predictive maintenance | AI can be used to predict when military equipment, such as ground vehicles and/or aircraft will require maintenance. This reduces downtime and ensures equipment is ready for deployment when needed. |
| Simulation and training | AI-driven simulations can provide realistic training scenarios for military personnel, helping them prepare for various situations from logistics to combat. |

Adoption of AI in the defense industry has been a topic of significant interest and debate. The use of AI in the military raises significant ethical, legal, and policy considerations. Concerns include the potential for autonomous weapons to make life-and-death decisions without human control, the potential for bias in AI systems, and the need for adherence to international laws and norms governing armed conflict. Military organizations worldwide are actively working to address these challenges as AI continues to play a growing role in both offensive and defensive operations.

Prior to developing the AI system, it is important to have a clear AI strategy that is linked to the prospective AI use cases. For example, a target identification application for identifying enemy personnel is a different use case than identifying enemy vehicles. Each use case has its own objective, data requirements, and ethical guardrails.

While there may be many approaches to identifying use cases, an AI use case template can aid in working toward a comprehensive, thoughtful AI strategy (Marr, n.d.). Table 5 provides descriptions for the template sections to complete for each use case.

**Table 5. AI use case template**

| Template item | Supporting questions |
|---|---|
| 1. Link to strategic goal | What is the strategic goal that this AI use case will support? |
| 2. Objective | What are the objectives of this AI use case? |
| 3. Measures of success | How will success be measured? What metrics will be used to track progress? |
| 4. Use case owner | Who will be the owner or sponsor of this AI use case? |
| 5. AI approach and required data | What AI approach will be used and what data will be required? |
| 6. Ethical and legal Issues | Are there any ethical or legal issues regarding this use case? |
| 7. Technology and infrastructure | What are the technology and infrastructure challenges and requirements? What new hardware and software is required? |
| 8. Skills and capacity | What are the challenges around skills, capabilities, and resourcing? How will any gaps be managed? |
| 9. Implementation | What are the implementation challenges? Who will deliver the project? |
| 10. Change movement | How will the project impact current operations and processes? How will any changes to those operations be managed? |

Once the template is complete for each use case, the developer can identify common themes, issues, and requirements. This approach will help ensure that AI is developed and applied within the bounds set forth by the use cases. Since AI systems may be used for different tasks, understanding the specific use cases for which the system will be employed will prevent misapplication of the AI system.

## 4.2 Requirements

AI reliability is critical for ensuring that AI systems can be trusted and used effectively in various applications. To achieve AI reliability, several requirements need to be met. These requirements encompass various aspects, including data, model development, testing, and deployment. Table 6 contains some of the key AI requirements to help ensure AI reliability, robustness, and resilience.

**Table 6. AI system requirement categories**

| Requirements | Description (with selected examples/scenarios) |
|---|---|
| High-quality data sets | Clean and representative data sets ensure that training data is free from biases, errors, and outliers and is representative of the real-world conditions the AI system will encounter. In addition, collect enough data to train a robust and generalizable model. |
| Testing and validation | Conduct rigorous and comprehensive testing, including system testing, integration testing, and end-to-end testing, to identify and fix issues. Additionally, test AI systems against adversarial attacks to assess their robustness. |
| Model development | Model development must be robust in selecting appropriate ML or deep learning algorithms that are suitable for the problem domain and design robust model architectures that can handle different input variations and perform well in various mission profiles or scenarios. |
| Error handling | Implement protocols/mechanisms that allow AI systems to degrade gracefully when faced with unexpected situations or errors. Furthermore, develop contingency plans and fail-safe procedures to minimize the impact of AI failures. |
| Documentation | Maintain thorough documentation of AI systems to include data sources, model architectures, and version histories. In addition, ensure compliance with relevant regulations and industry standards (e.g., ISO/IEC 27001 [ISO/IEC, 2022b], and DOD Directive 3000.09 [DOD, 2023]). |
| Security | Implement robust security measures to protect AI systems from hacking, cyberattacks, and data breeches, and control access to AI systems and data to prevent unauthorized usage. |
| Bias mitigation | Mitigate and evaluate biases in the training data and model predictions to ensure fairness and equity in all AI applications. In addition, develop AI systems with ethical principles in mind, addressing issues related to fairness, privacy, and transparency. |
| Model retraining | Periodic updates of training data to account for changing real-world conditions are critical to maintain model reliability and performance. |

This list of requirements is not meant to be an exhaustive list for ensuring AI reliability. It is an ongoing process that involves collaboration among data scientists, engineers, domain subject-matter experts, and stakeholders throughout the AI system's lifecycle. It requires a commitment to quality, transparency, and ethical considerations to build trustworthy and reliable AI systems.

A potentially powerful approach to aid in developing and refining system requirements uses a systems-view of the AI system. The systems-view considers the complex interactions between the human operator, physical hardware, software, AI algorithm, and any autonomous controllers. System-Theoretic Process Analysis (STPA) is an approach that employs such a systems-view to further understand complex systems. STPA is a hazard analysis technique based on an accident causation model. It examines potential losses and hazards by analyzing a system's control structure. A sample control structure for an autonomous system is shown in Figure 1. In this basic control structure, the downward arrows illustrate control actions, and the upward arrows

illustrate feedback paths. In general, control actions are based on the mission and feedback received.

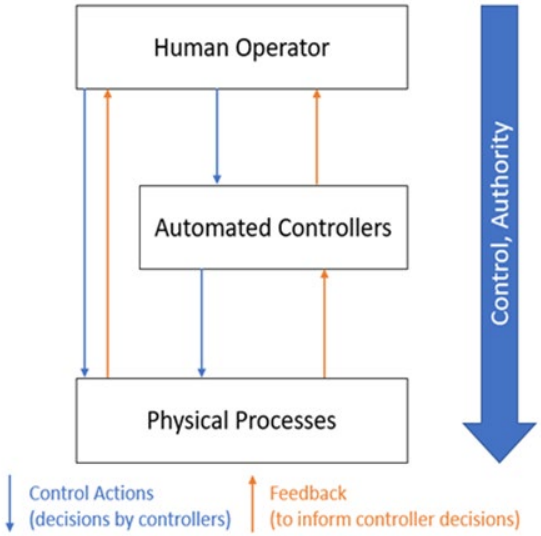


**Figure 1. Sample control structure of an autonomous system**

STPA assumes that accidents, or undesirable outcomes, are caused not only by component failures, but also through unsafe interactions between fully functioning components (Levenson & Thomas, 2018). AI and autonomous systems are ideal candidates for STPA, as the autonomy introduces a new level of complexity. For example, in the sample control structure, both the human operator and automated controller are controlling the physical processes. What if there is a conflict in control actions between the two? What if different feedback is presented to each? How does the system resolve this conflict? STPA provides a methodical approach to work through undesirable control actions and the conditions or scenarios that can cause them. STPA can help identify system requirements and constraints that will help mitigate potential failure modes and undesirable interactions between system components. Furthermore, it can be used to identify potential attack vectors, since every control action is an opportunity for manipulation or compromise of the system.

## 4.3 AI Model and Data

AI systems continue to rely on the data pipeline when put into production, making it an integral part of the AI system. Classes of data pipeline failure modes include performance bias failures, out-of-distribution (distributional shifts) failures, and cascading model failures (Blood et al., 2023). There are three types of data needed to ensure that the AI system is functioning properly and to lessen risk of these failure modes:

1. Training data – initial data used to develop the model.
2. Validation data – data that assesses and informs the choice of algorithm and parameters of the model. This data provides an opportunity to finetune the parameters during the model's development.
3. Testing data – data that the model has not seen, which provide an understanding of the model's performance on new examples. This data should be unlabeled.

It is imperative to have a constant refinement and optimization of training data as real-world conditions evolve (Rizzoli, 2022). Table 7 depicts the four types of learning methods that determine the type of training data that could be used for the model.

**Table 7. Learning methods**

| Learning method | Description |
|---|---|
| Supervised learning | The human operator labels raw data to inform the model about what it needs to learn. This places emphasis on the human-in-the-loop process as human operators can provide continuous feedback to improve the model's performance. |
| Unsupervised learning | The model finds patterns within the raw unlabeled data. For instance, recognizing how similar/different two data points are based on common features extracted. |
| Semi-supervised learning | A combination of supervised and unsupervised learning, where the data are partly labeled, and some predictions are left to the model's judgment. This is typically done when the model can be directed toward an area of focus, but actual predictions are hard to annotate as they are too small and/or nuanced. Most learning will fall in this category. |
| Reinforcement learning | Based on rewarding desired behaviors and/or punishing undesired ones. |

### 4.3.1 Data Quality

As alluded to previously, quality training data requires a significant amount of data cleaning and data labelling. Data labelling involves identification of raw data and at least one additional label to specify context so the model can make accurate predictions (IBM, n.d.). Labeled data may be more difficult to acquire and store, but it can help determine actionable insights for the system. It is good practice to continually look for ways to improve the label/data consistency and quality as the AI model is trained and deployed. The data must be "taken from a diverse set of sources under varied circumstances and then labeled and presented in a manner that the ML system can ingest" (Whitty, 2022). This means the data should have traits of accuracy, completeness, timeliness, relevancy, consistency, uniformity, and comprehensiveness. Moreover, the quality is affected by the people, process, and tools. Regular training (adjusted to the complexity of tasks) for workers, along with allowing room for iteration with data labelling and maintaining flexibility with tools, will be necessary to maximize quality.

### 4.3.2 Data Audits and Monitorability

As we go through the data pipeline, data audits can be performed to check whether data sets are reflective of the real world. They enhance the accuracy of the data and investigate the degree to which data sets are balanced, unbiased, applicable, and complete (Ammanath, 2022). They involve reviewing leading factors to make conclusions about the characteristics of a data set. Data audits will go hand in hand with monitoring the model in production. Monitorability will help deal with concerns of data quality and analyzing model performance. There are two levels of monitoring: functional and operational. Functional monitoring looks at the inputs, the model, and the outputs. At this level, the risk of failures like data drift or distributional shifts can be reduced with techniques such as using basic statistical metrics or tracking changes in distribution of real labels. Operational monitoring considers the whole AI system, the pipeline, and the costs (Oladele, 2023). Here, the system performance metrics become a key factor in ensuring the system is functioning properly. In addition, monitoring must be done throughout the lifecycle of the system, not only while it is in production. Monitoring for data quality, model quality, software quality, and service quality may be needed, depending on where the system is in the lifecycle.

### 4.3.3 Data Governance

Data audits and monitorability are some procedures that would be included in a data governance framework. Data governance is the process of managing the availability, usability, integrity, and security of the data in enterprise systems, based on internal data standards and policies that also control data usage. Data governance usually has elements like data mapping/classification and data catalogs. It will ensure that the data is consistent and trustworthy. This process should include a framework that spells out things such as a mission statement for the program, its goals and how its success will be measured, as well as decision-making responsibilities and accountability for the various functions that will be part of the program. It should be documented and shared internally so it is clear to everyone involved upfront how the program will work (Stedman & Vaughan, 2022).

## 4.4 Human–Machine Interaction

Human–machine teaming is an important consideration when developing and deploying AI systems. Humans teaming with AI can take many forms. Humans can supervise an AI system that is serving as an aide or helper. This is like traditional non-AI systems, in that the human is in control of the AI while using it to complete mission tasks. Humans can also collaborate with an AI system as equal teammates. This collaboration allows for decision-making by the AI system or the human, depending on how the system is

designed. A final form is the AI acting as a limiter of human performance. This may occur when the AI system and the human have different assessments of the world driving their decisions, affecting the level of shared situational awareness (Endsley, 2017).

Failure modes possible within the human–machine teaming involve three main classes: cognitive load, misunderstanding, and trust. Cognitive load failures result from shifts in the cognitive load placed on the operator. Misunderstanding failures occur when human operators do not correctly assess the state of the AI. Trust failures occur when the operator's belief in the level of the AI system's performance is misplaced (Blood et al., 2023).

### 4.4.1 Communication

To reduce the occurrence of these failure modes and, in turn, have good teaming and trust, we need to consider three characteristics: communication, transparency, and explainability. First, there needs to be bidirectional communication that includes information about the automation's intent, decision, action, and capabilities. AI systems are often designed as components of larger systems. Interdependence exists between the human and machine and between the human–machine team and the larger network of connected teams and systems. The system needs to be capable of having a "shared focus" with human operators (National Academies of Sciences, Engineering, and Medicine, 2022). Shared mental models between the human and the AI system can have a strong effect on performance in terms of understanding each other's roles and predicted behaviors. Human–human teams benefit from efficient information sharing despite the chance of greater workload and misunderstandings. Hence, it is not farfetched to have human–machine teams reflect that sort of relationship.

In addition, the level of communication will depend on the complexity of the AI system. Implicit communication will depend on suitable human sensors and algorithmic inferences, while explicit communication needs to support the management of common ground without distracting or disrupting the primary task (Ezer et al., 2019).

### 4.4.2 Transparency

Transparency includes communicating information about goals and proposed actions, reasoning processes behind those actions, and uncertainty associated with those actions. Higher transparency leads to an increase in team situation awareness, performance, trust, and perceived understanding of the AI system. It can also decrease time, pressure, and frustration within the team. However, there must be a balance, as it can also reduce the understanding of users if the provided information is confusing.

*IEEE P7001: A Proposed Standard on Transparency* presents a definition of transparency levels that could be applied differently depending on the stakeholders involved (Winfield et al., 2021). Level 1 represents what may be found in well-designed present-day systems or what can be easily provided. Levels 2 and up are successively more demanding. Table 8 shows a proposed table depicting transparency levels for end users for robots (Winfield et al., 2021).

**Table 8. Sample transparency levels**

| Transparency levels (Non-cumulative) | Definition |
|---|---|
| 0 | None |
| 1 | A user manual must be provided, which sets out how a robot will behave in different circumstances. |
| 2 | The user manual should be presented as an interactive visualization or simulation. |
| 3 | The robot should be equipped with a "why did you just do that?" function which, when activated, provides the user with an explanation of its previous action, either as displayed or spoken text (Koeman et al.,2020). |
| 4 | The robot should be equipped with a "what would you do if …?" function. |

In Table 7, the levels are mutually exclusive, meaning that we could have Level 2 and not Level 1 for the system. For most stakeholder groups, each level will build on previous levels. This table provides an example of what transparency may look like for an AI system. Note, however, there would be no expectation that if a system meets a particular level in one stakeholder group, it should also meet the same level in other groups. Moreover, this implies transparency must be examined at every point in the system's lifecycle. The model must be transparent enough to help with test and validation as well as allow the end user the option to remedy an unfavorable decision made by the model.

### 4.4.3 Explainability

As a part of transparency, explainability is an important element to have for an AI system. An explainable system should enable humans to understand rationale behind a decision in context. It should also allow them to accurately predict the effectiveness of AI, interpret an error, and decide what to do about it (Ezer et al., 2019). With this property, the AI system can potentially tackle issues with "black box modeling." Like transparency, explainability must be considered throughout the entire lifecycle. It is important to note that it will look different depending on what information is being communicated. For example, the model must be explainable enough for engineers to fix any bugs, which will be different from how the model will explain outcomes to end users. Explainability will depend on the monitoring practices implemented as there will be a

need for mechanisms to observe changes in the model versus changes in the system. Furthermore, an explainable AI system allows a user to revise their mental model, which can lead to better performance.

### 4.4.4 Trust

Trust is the determining factor of a productive collaboration. It is “the measure of strength in the expectancy that interactions with another entity will result in positive outcomes within an uncertain and risky environment” (Ezer et al., 2019). This measure must be a repeat measure and sensitive to negative trusting states, as there will always be a combination of justified and unjustified trust/mistrust in human–machine teaming. An appropriate level of trust requires the proper data hygiene (as described in the training/production section) along with the three previously discussed characteristics. Additionally, frequency of interactions, task handoffs, and accuracy of judgments on teammate behaviors are other indicators of trust levels. There needs to be a magnitude of security, integrity, assessment, and adaptability, as well as training of human operators with the system to achieve appropriate trust of the AI system. Trust will build over time as the human and AI interact often. Trust must be considered at all stages, from guiding the design to the production and deployment of the system.

## 4.5 Simulation for AI and Autonomous Systems

As with any other type of system, AI/AA software needs to be tested and evaluated before it can be fielded. For the reliability side of testing, these systems can require millions of hours or miles to show that they can be reliable for only a single environment. Due to the time and cost constraints, other methods have been deemed necessary to aid in testing the software, one of which being simulations for the AI system.

### 4.5.1 Simulation Overview and Design Choices

Simulations allow AI software to control the corresponding system in a virtual environment. These simulations can be easily repeated to gain insight on the AI systems much quicker than exclusively field trials would allow. Simulations can be used much earlier in the development process than field trials can, allowing faults to be identified and fixed earlier. Additionally, through self-teaching methods such as ML, AI can change and evolve very quickly through simulations to create a more reliable system. Simulations also allow us to expand the possible environments and scenarios the system can test within, allowing for a more robust software. Using simulations in tandem with field trials may lead to improved reliability of the AI software (Fremont et al., 2020).

To accurately test AI software for reliability using simulations, there are some key factors that must be considered. First, any simulated environment must be robust and realistic for the scenarios the AI will be placed into and should ideally test for resiliency in these environments as well. Being able to analyze the robustness and resiliency of the software is one of the key driving factors to use simulations for these tests. To accurately simulate AI software, the physics-related aspects should be dynamic and based on real-world scenarios and environments to ensure that simulations are able to match the results of what could be found in field trials. The simulations should also note the exact failure mode of the software to allow further analysis of any failures, such as those listed in Section 3. Using these failure types and identifying potential causes though the simulation is key to improving the reliability of the system (Taves et al., 2020).

At the start of any simulation-based test, engineers will need to understand the level of control the AI is designed to have. Some AI software will only be used to aid an operator and their decisions. In this case, the AI should be simulated for those situations, possibly with an operator involved. Other AI software is designed to operate as part of a larger system, where it has some level of command of that system. In these cases, the simulations would need to include this control aspect.

A more important decision involves how the results of these simulations will be used to improve the software. AI systems have the capability to teach themselves based on the simulations they run through, and with each failure they can train themselves to be more effective in the given environment. Simulations allow self-learning to happen alongside testing, which can speed up the time it would take to have the system learn and change naturally through just field trials. This ML is one of the reasons why reliability testing is so difficult. The software is constantly changing how it functions, which leads to a greater need for these tests. Of course, it may not be enough to exclusively let the AI software train itself, and its developer may need to analyze the failure data for themselves. This would allow them to see how effective the training is and make changes to the software outside of the simulations. Part of this choice boils down to choosing a learning method for the AI system, although this choice can change as development on the algorithm progresses. These tests need to strike the correct balance for how much the AI system can learn on its own, with the developer accurately judging the most effective training path forward (Goodin et al., 2022).

### 4.5.2 Simulation Types

There are several types of simulations applicable to AI software testing that allow a variety of training methods and results feedback. As a baseline, 3D simulation software is a type that contains the entire virtual environment. These would use synthetic data to

create new scenarios to train the AI, which is very effective at simulating autonomous vehicles (AVs).

At the next level, real images and data from sensors or field tests are incorporated in the simulation. These allow more realistic tests to be done using real scenarios for the AI software's decision-making process and its ability to process the inputs it is given. These methods are faster since no physical testing is needed, but the developer may lack important information otherwise obtained from physical testing (Fremont et al., 2020).

Another simulation method that shows promise is the digital twin used for parallel systems. This simulation method uses a digital twin in a virtual environment alongside a matching physical environment in the real world, called the parallel system. These scenarios are linked to match up with things such as location, terrain, and weather. With this method, a real vehicle in the real world can impact the virtual simulated environment by gathering data and then adding scenarios to the virtual environment, such as obstacles or other vehicles. It allows the testing of object detection and training the system's response to important stimuli. By using this method to analyze the systems alongside one another, a more accurate analysis can be completed to see how the AI software works in the simulation using real inputs from a real vehicle. Overall, using the parallel system for the real world helps to test the dynamics of the system and gather data to use for the digital twin, while the virtual environment of the digital twin helps to test the AI software and its outputs for the scenarios based on the parallel systems inputs (Meng et al., 2022). Table 9 contains a list of several of the existing simulation software.

**Table 9. AI simulation software packages**

| Developer | Software | Description |
|---|---|---|
| Tech-X | VSIM | Advanced physics simulator, most applicable to network simulation. |
| Advanced Science and Automation Corp | IVRESS | 3D simulation software capable of testing AVs. |
| Siemens | Simcenter | 3D simulation software with an entire section devoted to autonomous testing software, as well as hardware to record real-time data from an AV. |
| Dassault Systemes | Simulia | 3D simulation software, capable of developing and testing automation. |
| Speedgoat | Full vehicle simulation | Allows full digital twins of vehicles with advanced physics to monitor vehicle condition. Used by the University of Munich's robotic race team for autonomous cars. |
| rFpro | Simulation software | Extremely accurate simulation software, capable of integrating sensors to test driver assistance systems with AI-controlled traffic. |
| SCANeR | AVSimulation | RADAR, LiDAR, camera-enabled AV simulation software. Complete package for designing and testing AVs. |

## 4.6 AI System Test and Evaluation (T&E)

T&E is most effective when integrated throughout all phases of an AI system's lifecycle. Setting clear intentions and learning goals around testing can help teams prioritize resources to ensure the testing is producing actionable information. Interpreting these results may require a diverse team with expertise spanning different content areas. Such areas include interpreting the meaning of increased uncertainty, recognizing the emergence of new classes of data, and determining the efficacy in human–machine teaming. Establishing metrics that are measurable throughout development and deployment can be used to indicate maturity, performance, and reliability.

### 4.6.1 T&E Strategy

Due to the complexity of AI systems and the dynamic environments they may encounter, it is infeasible to test exhaustively in every possible scenario. Therefore, it is helpful to decompose the system to understand the areas we can test more rigorously. Three non-distinct areas are (1) hardware and software, (2) human–machine, and (3) AI algorithm with data pipeline.

For the hardware and software, the T&E strategies for traditional non-AI systems can be applied. Hardware testing can expose the system to environments and stresses that the system is expected to see. Surfacing failures and implementing corrective actions will improve the reliability of the system. After all, how valuable is the most reliable AI if the hardware it operates on cannot stay operational? For the software, there are many traditional software testing methods. Some methods are made to identify bugs in the software. Other methods can identify logic errors in software (IEEE, 2016). Both forms of testing can identify failures and drive corrective actions.

Human–machine interface testing can identify potential shortfalls existing in the physical interaction within the system. Whether the user is interacting with physical hardware or a graphical user interface, testing these with traditional non-AI methods and techniques is certainly applicable. The other part of the human–machine interface testing pertains to the information flow between the user and AI system. This testing can help address shortfalls in transparency and trust. When AI is making decisions or taking actions, there should be an indication to the user. Feedback directly from the user during testing can indicate the level of transparency and trust in the human–machine team.

The final area of testing involves the AI model. While the AI model is trained on training data, testing the algorithm introduces new data. It is important to understand how the AI system classifies, labels, and uses this new data. Testing allows a thorough exploration of the model's behavior prior to being deployed in production. However, coverage of

testing in the context of AI systems is limited due to the singular nature of their data points. An AI system may perform as expected given one input, then behave in an unintended manner on a similar input. To help narrow the focus of testing, it is helpful to examine possible sources of uncertainty within the model. It is also helpful to assess risk levels in different content areas.

### 4.6.2 AI System Metrics and Measurement

A typical reliability metric for traditional non-AI systems is mean time between failure (MTBF). A failure is usually tied to a hardware or software anomaly. In the introduction, a broader failure definition is presented—an event when the AI, algorithm, or autonomous system does not perform as designed, including incorrect prediction or statistically different prediction for a given instance or inability to guard against an adversary. The “not perform as designed” seems to align well with traditional types of failures. However, “incorrect prediction” types of failures begin to include areas of performance, such as accuracy. The accuracy may be degrading because the model is not robust enough or able to handle changing environments or data inputs. The “inability to guard against an adversary” type of failure relates to the resiliency of the system.

AI system reliability assessments can use metrics in addition to MTBF to determine the likelihood of successful mission completion. Measures of accuracy, resiliency, and robustness can all be useful. More research into applicable metrics in these areas will contribute to a more comprehensive assessment of mission reliability.

Safety measures are an important part of any AI system assessment. For example, in an AV, a key design element in making safe decisions is the computing time. The ability for a system to identify its surroundings and quickly make corrections or respond to other external factors is vital to its performance in the field. The computing time for making decisions would also apply to the total number of decisions it makes over a given time frame. This is important since if an operator believes it is making too few decisions, then they may not have much trust in the AV. A safety score that uses the AV’s computing time over other design elements is a potential path forward for AV analysis and testing. The safety score is based on a Responsibility Sensitive Safety model, which denotes appropriate distances between an AV and its surroundings. The sooner an AV knows to make the adjustment, the better the safety score, which is why the computing time is so important. Some simulation methods could be used to help calculate these safety scores, since the decision-making time would be easily identified through them (Zhao et al., 2020).

# 5. CONCLUSION

AI and autonomy applications are becoming more prevalent in military technologies. The reliability of these systems, or likelihood of mission completion, is now dependent on more than just the absence of traditional hardware and software failures. Data pipeline failures, human–machine failures, and intentional adversarial failures can also hinder mission completion. Several elements were presented that are important to decreasing the risk of fielding an unreliable product. Elements like use cases and requirements can help focus developmental efforts. Data pipeline and human–machine interaction elements can help ensure quality and transparency of AI system operations. Finally, simulation and T&E are useful for maturing and measuring AI products.

The elements presented will form the basis for DAC's development of a reliability risk assessment methodology and tool. Such a tool can be used by the S&T and acquisition communities to inform and help ensure mission reliability in new AI systems.

# 6. REFERENCES


Ammanath, B. (2022, March). *7 techniques for building reliable AI models.* Future. https://future.com/7-techniques-for-building-reliable-ai-models/

Blood, J., Herbert, N., & Wayne, M. (2023, January 23–26). Reliability assurance for AI systems*. Proceedings of the 2023 Annual Reliability and Maintainability Symposium (RAMS),* Orlando, Florida, 1–6.

[CMU] Carnegie Mellon University. (2021). *Robust and secure AI.* CMU Software Engineering Institute.

[DOD] Department of Defense. (2022, June). *U.S. Department of Defense responsible artificial intelligence strategy and implementation pathway*. U.S. Department of Defense; 2022.

[DOD] Department of Defense. (2023, January 25). *DOD directive 3000.09. Autonomy in weapon systems.* U.S. Department of Defense; 2023.

Endsley, M. R. (2017). From here to autonomy: lessons learned from human–automation research. *Human Factors*, *59*(1), 5–27.

Ezer, N., Bruni, S., Cai, Y., Hepenstal, S. J., Miller, C. A., & Schmorrow, D. D. (2019, October 28–November 1). *Trust engineering for human–AI teams.* Proceedings of the Human Factors and Ergonomics Society 2019 Annual Meeting, Seattle, Washington. Human Factors and Ergonomics Society; c2019. *63*(1), 322–326. https://journals.sagepub.com/doi/pdf/10.1177/1071181319631264

Fremont, D. J., Kim, E., Vardhan Pant, Y., Seshia, S. A., Acharya, A., Bruso, X., Wells, P., Lemke, S., Lu, Q., & Mehta, S. (2020, September 20–23). *Formal scenario-based testing of autonomous vehicles: from simulation to the real world*. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece. 1–8. doi: 10.1109/ITSC45102.2020.9294368. https://ieeexplore.ieee.org/document/9294368

Goodin, C., Carruth, D. W., Dabbiru, L., Hudson, C. H., Cagle, L. D., Scherrer, N., Moore, M. N., & Jayakumar, P. (2022, June 6). *Simulation-based testing of autonomous ground vehicles*. SPIE Defense + Commercial Sensing, Orlando, Florida. https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12115/121150J/Simulation-based-testing-of-autonomous-ground-vehicles/10.1117/12.2620502.short?SSO=1&tab=ArticleLink

Headquarters, Department of the Army. (2022). *Test and evaluation policy* (U.S. Army Regulation 73-1) (2022 draft).

IBM. (n.d.). *What is data labeling?* IBM Technology Corp. Retrieved August 15, 2023, from https://www.ibm.com/topics/data-labeling

IEEE Std 1633–2016. (2016, September). *IEEE recommended practice on software reliability* (1633–2016).

[ISO/IEC] International Organization for Standardization and the International Electrotechnical Commission. (2022a). *Trustworthiness — vocabulary* (ISO/IEC TS 5723). https://www.iso.org/obp/ui/#iso:std:iso-iec:ts:5723:ed-1:v1:en:term:3.2.12

[ISO/IEC] International Organization for Standardization and the International Electrotechnical Commission. (2022b, October). *Information security, cybersecurity and privacy protection – information security management systems*. (ISO/IEC 27001:2022).

Koeman, V. J., Dennis, L. A., Webster, M., Fisher, M., and Hindriks, K. (2020). The "Why Did You Do that?" Button: answering why-questions for end users of robotic systems. In L. A. Dennis, R. H. Bordini, & Y. Lespérance (Eds.) *Engineering Multi-Agent Systems* (pp. 152–172). Cham: Springer International Publishing. doi:10.1007/978-3-030-51417-4_8

Kumar, R. S. S., & Snover, J. (2022, November 2). *Failure modes in machine learning*. https://learn.microsoft.com/en-us/security/engineering/failure-modes-in-machine-learning

Levenson, N., & Thomas, J. (2018, March). *STPA handbook.*

Marr, B. (n.d.). *How to define your AI use cases – with handy template*. Retrieved August 15, 2023, from https://bernardmarr.com/how-to-define-your-ai-artificial-intelligence-use-cases-with-handy-template/

Meng, Z., Zhao, S., Chen, H., Hu, M., Tang, Y., & Song, Y. (2022). The vehicle testing based on digital twins theory for autonomous vehicles. *IEEE Journal of Radio Frequency Identification*, *6*, 710–714. doi: 10.1109/JRFID.2022.3211565. https://ieeexplore.ieee.org/document/9910602

Narayanan, S., & Carrier, R. (2021). *Accuracy, validity, reliability, robustness and resilience (AVR3)*. https://forhumanity.center/bok/accuracy-validity-reliability-robustness-and-resilience-avr3/

National Academies of Sciences, Engineering, and Medicine. (2022). *Human–AI teaming: state-of-the-art and research needs*. The National Academies Press. https://nap.nationalacademies.org/read/26355/chapter/7

Oladele, S. (2023, September 08). *A comprehensive guide on how to monitor your models in production*. Neptune.ai. https://neptune.ai/blog/how-to-monitor-your-models-in-production-guide

Rizzoli, A. (2022, July 11). *An introductory guide to quality training data for machine learning*. V7. https://www.v7labs.com/blog/quality-training-data-for-machine-learning-guide

Stedman, C., & Vaughan, J. (2022). *What is data governance and why does it matter?* TechTarget. https://www.techtarget.com/searchdatamanagement/definition/data-governance?Offer=abt_pubpro_AI-Insider

Taves, J., Elmquist, A., Young, A., Serban, R., & Negrut, D. (2020). *SynChrono: a scalable, physics-based simulation platform for testing groups of autonomous vehicles and/or robots*. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, Nevada. 2251–2256, doi: 10.1109/IROS45743.2020.9341585. https://ieeexplore.ieee.org/document/9341585

Whitty, R. (2022, June 8). The battle with data. Realities of bringing artificial intelligence to the battlefield. *Expeditions with MCUP (Digital Journal).* Marine Corps University Press. https://doi.org/10.36304/ExpwMCUP.2022.09

Winfield, A. F. T., Booth, S., Dennis, L. A., Egawa, T., Hastie, H., Jacobs, N., Muttram, R. I., Olszewska, J. I., Rajabiyazdi, F., Theodorou, A., et al. (2021, July 26). IEEE P7001: a proposed standard on transparency. *Frontiers in Robotics and AI, 8*. https://doi.org/10.3389/frobt.2021.665729

Zhao, H., Zhang, Y., Meng, P., Shi, H., Li, L., Lou, T., & Zhao, J. (2020, October 19–November 13). *Safety score: a quantitative approach to guiding safety-aware autonomous vehicle computing system design*. 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, Nevada. https://cseweb.ucsd.edu/~jzhao/files/safety-score-IV-2020.pdf

# LIST OF ACRONYMS

| | |
|---|---|
| 3D | three-dimensional |
| AA | assistive automation |
| AI | Artificial intelligence |
| AV | autonomous vehicle |
| CMU | Carnegie Mellon University |
| DAC | DEVCOM Analysis Center |
| DEVCOM | U.S. Army Combat Capabilities Development Command |
| DOD | Department of Defense |
| IEEE | Institute of Electrical and Electronics Engineers |
| ISO | International Organization for Standardization |
| LiDAR | light detection and ranging |
| ML | machine learning |
| MTBF | mean time between failure |
| RADAR | radio detection and ranging |
| S&T | science and technology |
| STPA | Systems-Theoretic Process Analysis |
| T&E | Test and Evaluation |

## DISTRIBUTION LIST

DEVCOM Analysis Center
FCDD-DAS-L/N. Herbert
FCDD-DAS-L/B. Hum
FCDD-DAS-L/P. Theragene
FCDD-DAS-L/A. Marchioni
6896 Mauchly St.
Aberdeen Proving Ground, MD 21005-5071

DEVCOM Analysis Center
FCDD-DAD-TP/E. Chatterton
Redstone Arsenal
Huntsville, AL 35898

DEVCOM Army Research Laboratory
FCDD-RLB-CI/Tech Library
2800 Powder Mill Rd.
Adelphi, MD 20783

Defense Technical Information Center
ATTN: DTIC-O
8725 John J. Kingman Rd.
Fort Belvoir, VA 22060-6218