NPS-CS-22-003

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

Understanding, Assessing, and Mitigating Safety Risks in Artificial Intelligence Systems

Joshua A. Kroll and Valdis Berzins

December, 2022

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

Prepared for: Naval Air Warfare Development Center (NAVAIR). This research is supported by funding from the Naval Postgraduate School, Naval Research Program (PE 0605853N/2098). NRP Project ID: NPS-21-N387-A

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE **12/20/2022**	**2. REPORT TYPE** **Technical Report**	**3. DATES COVERED** **START DATE** **01/01/21**	 **END DATE** **12/31/21**

4. TITLE AND SUBTITLE

Understanding, Assessing, and Mitigating Safety Risks in Artificial Intelligence Systems

5a. CONTRACT NUMBER	**5b. GRANT NUMBER**	**5c. PROGRAM ELEMENT NUMBER** **0605853N/2098**
5d. PROJECT NUMBER **NPS-21-N387-A**	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)

Joshua A. Kroll and Valdis Berzins

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School** **Monterey, CA 93943**	**8. PERFORMING ORGANIZATION REPORT NUMBER** **NPS-CS-22-003**

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **Naval Air Warfare Development Center**	**10. SPONSOR/MONITOR'S ACRONYM(S)** **NAVAIR**	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)** **NPS-CS-22-003;** **NPS-21-N387-A**

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views expressed in this document are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

14. ABSTRACT

Traditional software safety techniques rely on validating software against a deductively defined *specification* of how the software should behave in particular situations. In the case of AI systems, specifications are often *implicit* or *inductively defined*. Data-driven methods are subject to sampling error since practical datasets cannot provide exhaustive coverage of all possible events in a real physical environment. Traditional software verification and validation approaches may not apply directly to these novel systems, complicating the operation of systems safety analysis (such as implemented in MIL-STD 882). However, AI offers advanced capabilities, and it is desirable to ensure the safety of systems that rely on these capabilities. When AI tech is deployed in a weapon system, robot, or planning system, unwanted events are possible. Several techniques can support the evaluation process for understanding the nature and likelihood of unwanted events in AI systems and making risk decisions on naval employment. This research considers the state of the art, evaluating which ones are most likely to be employable, usable, and correct. Techniques include software analysis, simulation environments, and mathematical determinations.

15. SUBJECT TERMS

Artificial Intelligence, Safety, Assessment, Evaluation, MIL-STD 882

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**
a. REPORT **U**	**b. ABSTRACT** **U**	**C. THIS PAGE** **U**	**UU**	**67**

19a. NAME OF RESPONSIBLE PERSON Joshua A. Kroll	**19b. PHONE NUMBER** *(Include area code)* **(831) 656-3988**

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Ann E. Rondeau
President

Scott Gartner
Provost

The report entitled “Understanding, Assessing, and Mitigating Safety Risks in Artificial Intelligence Systems” was prepared for the Naval Air Warfare Development Center and funded by the Naval Postgraduate School, Naval Research Program (PE 0605853N/2098).

This report was prepared by:

Joshua A. Kroll, Assistant Professor
Computer Science Department

Valdis Berzins, Professor
Computer Science Department

Reviewed by:

Gurminder Singh, Chairman
Computer Science Department

Released by:

Kevin B. Smith
Vice Provost for Research

THIS PAGE INTENTIONALLY LEFT BLANK

EXTENDED ABSTRACT

Traditional software safety techniques rely on validating software against a deductively defined *specification* of how the software should behave in particular situations. In the case of AI systems, however, specifications are often *implicit* (e.g., in the case of expert systems and rules engines, where behavior is induced by describing facts and deduction rules and allowing an inference engine to recombine them according to pre-set combination schemas) or *inductively defined* (e.g., in machine learning systems, the goal is to identify and repeat as predictions patterns in aggregated data sets, and the specification is implicitly derived from the data set and perhaps some parameters of the model describing or extracting the patterns). Methods that extract insights from data-driven analytics are similarly inductively defined and are subject to sampling error since practical datasets cannot provide exhaustive coverage of all possible events in a real physical environment, which usually has an unbounded set of possibilities. Thus, traditional software verification and validation approaches may not apply directly to these novel systems, complicating the operation of systems safety analysis (such as implemented in MIL-STD 882E). However, AI offers advanced capabilities, and it is desirable to ensure the safety of systems that rely on these capabilities. When AI tech is deployed in a weapon system, robot, or planning system, unwanted events are possible. There are several techniques that can be used to support the evaluation process for understanding the nature and likelihood of unwanted events in AI systems and making risk decisions on naval employment. This research considers several of those techniques, and evaluates which ones are most likely to be employable, usable, and correct. Techniques include software analysis, simulation environments, and mathematical determinations.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION

Artificial Intelligence (AI), a suite of technologies which widen the scope of tasks that can be automated productively, has captured the attention of the defense community around the globe. AI creates opportunities for automation in applications that were previously thought to require close human supervision, offering advanced capabilities and new efficiencies. Yet important questions remain:

- If we delegate tasks thought to require careful human oversight to machines, do we jeopardize safety and effective control?
- How can we test and evaluate technologies that stretch or exceed previously understood limits of functionality?
- How do humans navigate the interface with machines to build teams more capable than purely human efforts or purely automated tools?
- If AI is fundamentally epistemically limited (as all models and computational tools are) by expressing only rigid rule-driven behavior (perhaps stochastically), how can we assess performance in an open world, as practical data sets are necessarily limited?
- What can be done to manage ethical and policy problems attendant to novel automation?

Because these questions do not have straightforward or widely accepted answers, AI complicates the operation of standard system functionality and safety assessments, such as the acceptance criteria in the DoD Directive 5000-series processes or the operation of MIL-STD 882E-driven analysis.

Our work organizes what is known about these questions in service of building concrete safety and assessment frameworks for AI systems. We determine what existing methods do and do not establish with respect to the requirements of AI assessments, describe areas where existing techniques do not meet those requirements as well as what would be needed to cross the gap, and propose techniques and practices for applying emerging tools to the problems of safety of and for AI.

A. CHALLENGES IN VERIFYING SAFETY FOR AI SYSTEMS

Traditional software safety techniques rely on verifying software against a deductively defined specification of how the software should behave in particular situations. Specifications are defined either extensionally, by defining the input/output relation of a program, or intensionally, by defining the rules or methods of computation a program must use. Verification, then, is the problem of comparing the behavior of the real system to the specification. The complementary problem of validation aims to establish that the specification captures goals and intentions of system stakeholders.

However, in the case of AI systems, specifications are often implicit or inductively defined. In the case of expert systems and rules engines, the specification is implicitly provided by a set of facts and deduction rules, and behavior is induced by allowing an inference engine to recombine them according to pre-set combination schemas. In machine learning systems, the goal is to identify patterns in the data and to use then to make predictions about situations that have not yet been observed. The specifications for these systems are inductively defined by choosing a data set and the parameters of the training objective used to derive the model from the data. Although models such as linear and higher-order programs may include an explicit objective function, this does not specify what behavior is recommended or effected beyond that it must comport with optimal loss according to that objective. In both cases the models are complex, and it is difficult to draw general conclusions about their associated behavior, which may have unintended aspects [1].

Rather than defining behaviors directly or even providing rules for behaviors, AI systems structure possible behaviors and then use related desiderata and their structure to determine behaviors automatically. Methods that extract insights from data-driven analytics are similarly inductively defined, since specifying what is to be counted in an analytic or even setting thresholds on an analytic for when actions should be taken subordinates behavioral outcomes to the measured values in data sets. Thus, traditional software verification and validation approaches may not apply directly to these systems. For example, without an explicit specification, what does one validate or verify against? Emerging techniques could bridge portions of this gap, for example by defining and bounding the regions in which optimal solutions are sought, which enables strong claims of the form "any inductively defined behavior within the specified bounds will be safe" subject to the bounds established and the meaning of "safe" for that system. However, even such strong claims may not be enough to guarantee safe operation, as the claims rest on assumptions that are difficult to validate and often sit on shifting ground [2].

B. HIDDEN SUBJECTIVITY OF DATA-DRIVEN DECISIONS

Often, data-driven decision-making is cited as the paradigmatic ideal of efficiency and objectivity. But data are subject to biases that may be introduced by choice of sampling methods or other systematic measurement errors. Data-driven modeling thus requires assumptions about both how data do and do not reflect reality and about the robustness of generalization that is possible from a finite set of assumptions and data instances. We consider how to analyze the validity of these assumptions using approaches drawn from statistics, the emerging field of data and model bias analysis, and the established fields of construct validity and reliability. Data bias and its management are particularly relevant to ethical and policy questions around data-driven AI techniques such as machine learning.

Data-driven systems are limited epistemically: because data sets necessarily depict a finite, limited representation of the world, performance in real scenarios is necessarily lower than performance estimates produced in laboratory or test and evaluation settings. Even stipulating the impossible claim that available data perfectly operationalize the desired target variable, practical datasets cannot provide exhaustive coverage of all possible events in a real physical environment, which often has an unbounded set of possibilities. Safety concerns include the following questions:

- By how much does performance degrade for real-world data feeds, and can we bound this deterioration either over time or across deployment scenarios?
- Can this loss of performance be caused by an adversary?
- To what extent are epistemic guarantees subject to time bounds (that is, how do we know when data sets are out of date due to drift of real-world characteristics or possible sudden disruptive events)?

We review in Section IV, specifically drift and its management in Section IV.B.1 and robustness to generalization and adversarial perturbation in Section IV.G the framing of these problems and work to date on solutions to separate the fundamental limitations from the issues that can be overcome through careful design and analysis.

C. EFFECT OF HUMAN SUPERVISION ON AI SAFETY

Human intervention is often offered as a solution for mitigating adverse AI behaviors. The thinking is that if a human oversees the system ("human-on-the-loop") or makes decisions that lead to practical effects based on the recommendation of AI ("human-in-the-loop"), the human will correct misbehaviors and emergent activities of the automation. However, this approach discounts much knowledge in the domain of human-machine interaction and the vast literature on safety system effectiveness, which show that human supervisors of automated systems are often ineffective due to problems such as interface/mode confusion, automation dependency, alarm or vigilance fatigue, and deterioration of human skills due to lack of practice.

These automation bias problems are manageable, but their management requires an understanding of how they manifest in different types of AI systems and what is known about quantifying, qualifying, and responding to them. Existing work on "explainable" or "interpretable" AI supports such a response, but often focuses too narrowly on introspection about the behavior of a model, ignoring the broader epistemic trust questions raised above. In Section IV.C, we outline problems that can occur in human-machine teams and can increase the risks of adverse behaviors or undesired events. We also review briefly the extensive work on mitigation and management of these issues, focusing our analysis on the specific problems encountered in military AI applications and the ways in which expert human-factors engagement can improve the risk and safety profile of AI.

D. BALANCING RISKS AND BENEFITS OF AI

AI offers advanced capabilities, driving its adoption. The US National Security Commission on AI argued forcefully that adopting AI must be a national military imperative to maintain competitive advantage [3]. If, however, we adopt AI that is fragile or misbehaves, that very adoption could lead to an erosion of competitive advantage [4]. It is desirable to ensure the safety and efficacy of systems that rely on these capabilities, and to adopt those systems that are safe and effective.

When AI technology is deployed in a weapon system, robot, or planning system, unwanted events are possible. Many techniques can assess the risk of unwanted system behaviors, so that appropriate mitigations can be applied in the final system. The National Institute of Standards and Technology is presently engaged in an effort to systematize AI risks and risk mitigations into an AI Risk Management Framework (AI RMF) [5]. Section IV describes a hazard analysis across our taxonomy of AI system architectures and design paradigms. This analysis allows us to examine what such a framework must cover and compare these requirements to existing NIST frameworks in cybersecurity and privacy to maximize the likelihood that the new AI RMF will provide a successful tool for system evaluation.

E. TRUST AND ASSURANCE CASES FOR AI

Safety and efficacy risks in AI systems give rise to limitations on the trustworthiness of those systems for particular purposes. These limitations can lead to situations where systems are pulled or barred from deployment in contexts where they could provide new capability or add enormous value. Systems can also be deployed but then ignored or disabled by operators wishing to maintain effective control commensurate with their responsibility for outcomes.

Trust is a key dimension both at the individual human-system interface level and at an organization-wide governance level. In Section IV.H, we consider how trust questions manifest organizationally by examining techniques that support oversight and review in AI system development and operation, supporting existing DoD policies such as Directive 3000.09 governing "Autonomy in Weapons Systems" [6], operative safety and acquisitions standards, and the DoD AI Principles. For example, the principle of traceability suggests a need to review both individual and organizational decision-making that leads to the structure of the system in general in addition to specific AI system outcomes [7].

Risk and performance analyses are important for AI systems not only because they enable safe and effective deployment of new and enhanced capabilities, but also because we cannot reap the benefits of new capabilities without understanding attendant limitations and possible means for mitigating attendant risks. Allies, partners, and the public at large demand these answers. At present, assurance cases for building justified confidence in AI systems are ad hoc. This report systematizes requirements for and approaches to developing these arguments.

II. A TAXONOMY OF AI SYSTEM ARCHITECTURES

We begin by describing the basic nature of AI system structure. To taxonomize AI systems, we offer a series of essential differentiations and dimensions along which systems vary from each other. Our goal is to enable generalization with sufficient view of the structure of these systems to perform our hazard analysis below; our taxonomy is a model and, like all models, is necessarily incomplete and ill-fitting in some cases. Below, we describe the essential questions of our taxonomy with some exemplary systems along the spectrum of answers available to each.

A. DATA-DRIVEN VS. MODEL-DRIVEN DESIGN

Development and evaluation of both AI systems and classical software systems depends on choices about how they represent the system's goals and functionality with respect to the rest of the world. These choices form the core *world model* espoused by the system. World models often involve explicit data sets but need not. For example, in traditional software systems, data is present implicitly – a world model informs requirements, the flow-down of requirements into design, and performance or acceptance testing, which includes choices like "what level of performance is acceptable?". Although all systems are based on choices about how to represent the world, the nature of those choices can differ.

In systems where data is an explicit part of the process (e.g., data science and analytics; machine learning; systems that make use of databases), decisions about the correct world model are often driven by choosing the data or defining its properties:

- What information will be gathered?
- By what sensing mechanism?
- How will categories be defined?
- What should be done when values are missing or lie outside defined ranges?
- If data are to be labeled by system outputs, how will labels be generated, when, and by whom?
- How long will data be retained and who should have access for what purposes?
- How quickly will data and categorizations become obsolete and how often will it be updated?

Validating the world model of a data-oriented system requires considering the *measurement properties* of that data [8]. Measurement properties define the relationship between the world model operationalized by choices in a system's design, implementation, or operation and the abstract construct that represents the system's goals or desired outcomes. For observable, physical properties such as length, measurement is straightforward: length can be operationalized by measuring against standard lengths using tools such as a ruler, tape, or LIDAR. Each has known problems which contribute to error in both systematic and random ways [9]. Some constructs are not directly observable, such as the risk a vehicle part will fail, but can be operationalized by methods such as comparing the distribution of past failures to observable features of the part of interest that are expected to have a causal connection to potential vehicle failures, such as the age and operational history of the part. Choices made about operationalization are important because they can introduce bias or blindness to certain outcomes in the world model of a system [10], as we discuss below in our hazard analysis. Regardless, data themselves are a model of the world, chosen by measurement and thus not a direct representation of reality. (That is, although the data themselves are not chosen, the measurement methods are, and thus the world model of the system as a whole is constructed of these choices.) The frameworks of construct validity and construct reliability provide strong grounding for understanding measurement issues in AI systems [10].

In other situations, the development of a system's world model is driven less by choices about data or measurement activities and more by choices about how the system is built or should function. Of course, all systems require both sorts of choices – the distinction here is about the frame in which the system is designed and the primacy of the classes of architectural decisions.

For example, an expert system that recommends next steps in diagnosing faults in a complex vehicle engine might be based on fault trees or maintenance manuals generated by experts. In such cases, data are still present – the manuals and fault trees reflect information gathered from experts. Yet in these cases, we think in terms of "model-driven engineering" where choices about the model (choices about how to relate faults to observable features and how to compose these into decision trees, for example) are more important than choices about data (for example, it is unlikely that the methodology for gathering expert knowledge affects the design of this system in detail – experts may disagree about how faults are expressed in observable features, but the set of faults the system must recognize and the set of observable features is likely quite stable across experts and data collection methods).

In understanding hazards, it is important to recognize that such systems still reflect measurement and modeling choices about the world and to look beyond the details of the system to see the world model beneath and validate it appropriately. Many rule-driven systems obscure their nature as models: natural language models often rely on grammars to break streams of language into their constituent parts, but these grammars cannot distinguish distinct linguistic structures that yield the same sequence of tokens even if the contextual meaning of the structures can be quite different [11]. To resolve these ambiguities, such systems often use belief *ontologies* that define concept categories and relationships. In focusing attention on the choice of parsing rules and ontology structures, these systems obscure foundational measurement questions such as how the language being parsed does or does not represent the constructs of interest to the system. (For example, most systems are applied in a specified context or deployed for a specified purpose, and the performance of the parsing rules on language matching that context or for achieving the system's purpose is what matters in validating that the rules and ontologies are acceptable. However, the fact that parsing rules/ontologies seem to be freestanding rules obscures the choices made in developing, evaluating, and accepting them – they model the world just as much as data-driven architectures do.)

It should be noted that model-driven architectures are more like traditional declarative/deductively defined engineered systems than data-driven architectures are.

While they do not have a full declarative specification, model-driven architectures consist of fully declared components and so are more easily amenable to assurance arguments of the standard verification and validation form (as the process of choosing, evaluating, and accepting model structures provides leverage for V&V). By contrast, data-driven systems inherit the uncertainty of their data and its measurement processes. While it is possible to declare what about the world to sense and how, the relationships defined by the fruits of this data gathering are necessarily left implicit. In a model-driven architecture, it is at least possible to define and limit these relationships by construction, while this presents challenges in many data-driven architectures.

One especially important gap between model-driven engineered systems and the validation of their world models arises in the management of model parameters and hyperparameters that represent choices made about the points in its operating space where the system's behavior changes. For example, the passing threshold for a class, the adjudication of which items belong in particular data classes, or the level of overall or subclass-disaggregated accuracy at which a system is deemed appropriate to use are all parameters that must be chosen. These same questions arise in data-driven system architectures, but their relevance to the system's performance is clearer in a relative sense, since data-driven system behavior is naturally mediated by parameter choices while in model-driven architectures these parameters become part of the modeling choices made. (That is, choosing a parameter is a modeling decision, while in a data-driven paradigm one might set the parameter either according to some feature of the data or simply as the parameter that optimizes the chosen objective.) An interesting (and open) line for future research is the extent to which developers of systems in different categories recognize and actively choose to manage the risk of validation gaps between their modeling choices (either in a data-driven or model-driven architecture) and the system's context of operation. That is, do developers notice gaps between the world model they are constructing and the reality their systems will eventually need to deal with? Mishap risk can depend on these parameter choices, and their management is tricky in all systems. We note these choices as especially difficult for model-driven system architectures, where the correct parameter value might depend on latent consequences of modeling decisions made implicitly during model-driven design. For example, standard grammars for language detection have been shown to fail when presented with novel dialects and tokenizing grammars might not perform equally well across different strata within language corpora [12].

Another issue that falls along the dimension of data-driven vs. model-driven design is the stochasticity of the resultant model. When models are overtly data-derived, questions like "what is the distribution of possible models available" are straightforwardly answered using bounds on measurement error and other sources of nondeterministic uncertainty. Epistemic error management is more challenging but possible through a combination of overt measurement modeling and bootstrapping of confidence or prediction bands for the generated model. We describe these issues further in Section IV.A.

Model-driven architectures, which are more amenable to verification and validation, may be stochastic in nature as well, either by virtue of stochasticity introduced during the modeling process (e.g., stochasticity resulting from the distribution of rules or parameters) or because deterministic response to stochastically defined inputs yields stochastic output (i.e., explicitly deciding to depend on a source of uncertainty – many software programs, for example, derive entropy from the low-order bits of their startup time, which may be useful to avoid complete determinism but also insufficient to provide security against an adversary who has a prior on the startup time or who can observe it). Here, determining the parameters of the distribution of outputs is much more challenging and may require new assumptions in the system's world model, such as distributional assumptions about inputs. It is important to make plain the dependence on distributional assumptions when validating a system's world model, since making unwarranted assumptions increases the distance between the model and reality, possibly introducing failure modes or increasing failure likelihood. Although many methods are general in the sense that they do not declare explicit distributional assumptions or appear structurally different when different assumptions are taken, world models generally presume some assumptions about the distributions of relevant environmental events, and claims of distribution-independence should be evaluated carefully. When coupled with real-world inputs, all models should be considered to have some stochastic component, even models that work on purely deterministic rules such as expert systems. A claim that the variance is small enough that stochasticity can be disregarded is an assurance claim that requires a supporting argument.

An additional set of world model gap concerns arises from the fact that the world is non-static, but once developed most systems are at least somewhat fixed in structure or behavior. The points of intervention for maintaining the currency of an AI system, and the ability to detect drift of a system's world model from underlying reality, depend on the AI system's architecture (e.g., data-driven vs. model-driven). Additionally, architectural choices can affect the rate of drift. Especially when production cycles are long or requirements determination is far separated from system delivery and use, it is natural to expect some gap between the system's established, mostly fixed world model and the changing state of reality. This can happen before or after validation or measurement modeling exercises, and managing drift must be an ongoing, active process. In some cases, this drift might even be affected by the system itself, a problem known as *endogeneity*. Feedback (positive or negative) based on the presence of the system can arise when the features a system measures are affected by its outputs. For example, a system that recommends diagnosis and debugging steps for a complex vehicle engine will lead technicians to perform maintenance activities according to the system's recommendations, which will affect the data used to develop or validate future systems or future versions of the extant system.[1] This feedback creates an implicit system-belief state that can drift away from the underlying reality. In the complex engine maintenance example, systematically following the maintenance recommendations can lead to more frequent maintenance activities in a self-reinforcing way (e.g., technicians perform the maintenance because it is recommended, but it is recommended because data show these are the steps technicians take prior to successful resolution of problems). This reinforcement can mask the performance of alternative models: even if less frequent

[1] A non-military example comes from real estate, where the prices people are willing to pay for homes are heavily influenced by freely available automated valuation models available on popular websites, but the prices reported by these models are heavily affected by the prices people actually paid for homes.

maintenance activities would not lead to more frequent engine failures, the system will not recommend this; even if a different set of debugging activities would lead to a quicker resolution, these paths remain unexplored because the system has affected the activities of the technicians. In general, claims about world model validation must be scoped to a period of time and require an argument about the extent to which concept drift and feedback affect validation. Such drift of system-belief states can sometimes result in rare but sudden and potentially catastrophic corrective transitions, such as stock market crashes, when the gap between belief-state and reality becomes large enough for the system to become unstable. We explore concrete mitigation strategies for managing drift in Section IV.B.1.

We find that those systems where architectures are primarily driven by choices about data make overall verification more challenging while making interventions that support validation easier. Policies against which to verify must be assumed in such systems, but also such systems are straightforwardly amenable to interrogation of their measurement properties and perhaps explicit measurement modeling. By contrast, when system architecture is driven by modeling choices, overall verification is more straightforward, but validation requires investigation into the latent choices driven implicitly by modeling decisions. It is clearer in such cases what to verify against (because there have been explicit choices), but assumptions about the system's world model may be obscured (by those same choices). In these cases, because interventions that reduce the epistemic gap between a system's world model and reality require modification to the system, it can be harder to correct problems when discovered.

As shown in our hazard analysis in section IV of this report, there is not a complete assurance story for either type of system at present; more research is required to fill in gaps in the AI assurance space.

B. DIRECT AUTOMATION VS. DECISION SUPPORT

Systems are often described by their degree of "autonomy". Some systems, such as those responsible for e-mail delivery, yield their intended effects directly without human intervention. Other systems serve in *decision support* roles, such as those highlighting certain features of an image presented to a medical radiologist and are designed to support a human decision-maker. Often, this distinction is reduced to the core question of where the locus of agency sits: is there a "human-in-the-loop" or not? Fears that placing a slow, fallible human between machine-generated reactions and real-world effects may introduce dangerous delays lead to designs where humans supervise direct-effect systems in a so-called "human-on-the-loop" architecture. All these categorizations can be understood as a spectrum of degree and nature of human involvement in intended outcomes.

Although it is tempting to suggest that systems which have humans involved in their chain of effect are more tightly controlled than systems that give direct effects, this is a gross oversimplification.

First, even when humans are involved in the causal chain leading to a system's effect, it is difficult to disentangle human agency and system structure; many failure analyses struggle with the question of whether an unintended event was a system failure or operator error. Humans may confidently follow the system even when it is wrong or may disregard unexpected system recommendations even when they are correct.

Second, it is clear on reflection that all AI systems involve humans to at least some extent: humans create the systems and direct them to a goal. Systems in operation are supervised for their fidelity to the goals of the operators. The operative question, therefore, is not *whether* or even *to what extent* humans are involved in the causal chain leading to system behaviors, but whether the humans involved can respond adequately when those behaviors are undesirable.

- Can humans override individual decisions?
- Do operator-level concerns factor in the system structure and architecture?
- Can they stop an unintended effect before it is reified by the system?
- Will humans notice when the system has entered a degraded or failed state, and can they rectify this?

The example of an e-mail delivery system makes this point plain: although the system is "autonomous" once it is operating, well controlled e-mail systems provide metrics to their operators about how much harmful mail is being blocked, how much overall mail is being delivered, and other relevant statistics that characterize the work of running a safe and functional e-mail service. In the case of an increase in failures, the operators and controllers of a system might change the structure (e.g., by purchasing a spam filtering appliance or blacklisting problematic sending servers), might change operational parameters (e.g., decreasing the threshold at which various mitigations to overwhelming traffic are applied), or might take follow-up actions to rectify system errors (e.g., removing phishing e-mails designed to steal end-user log-in credentials from those users inboxes after delivery). What matters is not whether humans are involved (they always are), but what *affordances* the system provides them in terms of an action space and situational awareness to respond to and (when necessary) override the system's behaviors.

Accepting that humans are always in the loop highlights the complications introduced by humans acting as system components.

First, as noted above, humans presented with a system's hints about their choice of action may be confused about the state of the world by those very hints, either not knowing how to interpret the information provided to support their decision or not knowing whether they should be confident enough to override the machine. Studies show that humans rate the quality of machine-provided advice as lower than human-provided advice regardless of its actual quality, but also follow that advice more often [13].

Second, humans may be confused about how to respond to machine guidance, either not understanding the state of the machine and how it will respond to their input or simply not understanding how to give feedback appropriate to the situation [14]. For example, in both the at-sea collision of the *USS John S. McCain* in 2017 and the loss of Air France Flight #447 in 2009, trained and experienced crews operating at a high level of safety discipline nevertheless misinterpreted signals from the systems they were operating in ways that led to fatal, catastrophic mishaps.

This illustrates the second complication of humans: as human power in a system is multiplied by automation, less human attention is of necessity spread over more system behavior. **This means that humans become less aware of system behavior and less familiar with it, even while they are more critical to the oversight of that behavior**, the so-called "paradox of automation" [15]. This makes humans involved in the operation of a safety-critical system less the locus of agency and more the putative locus of blame when the system fails, or what Elish calls "moral crumple zones" [16].

From a safety analysis perspective, it is thus important to determine not just if humans are involved in the system but to what end, and whether they have the power to understand and affect the behaviors of the system or whether they are held captive by its structure.

C. OPEN VS. CLOSED-WORLD SYSTEMS

Beyond the problem of measurement raised above, which relates the world-model of a system to reality, there is the problem of the horizon of any finite model, which must necessarily only describe a finite number of states of the world. The problem is that the real world has no such bound on possible states.

All automated systems will eventually encounter situations in operation that were not foreseen during development or evaluation. How well will such systems perform in these situations? The answer is generally that AI systems are fragile when asked to generalize, especially for situations that are not covered by available data and prior experience. We discuss this in more detail in the hazard analysis of Section IV.A and describe the capabilities and limitations of known mitigation techniques. However, the fundamental problem remains.

Thus, safety analysis must determine whether the system in operation will encounter a *closed world*, in which the number and kinds of outputs is naturally bounded (e.g., identifying known personnel allowed to enter a facility) or an *open world*, where the space of outputs the system might be called on to give is not limited a priori (e.g., matching characteristics of an individual – such as facial imagery – to the open ended question of identifying that person). For example, in the problem of classifying objects in an image, AI systems typically will be built to select among a set of predetermined categories. Say there are *n* such categories – in real operation, there is no assurance that the system will not see an object which does not fit into one of the existing categories or that it will not see at least $n + 1$ kinds of objects. Yet in either case such a system must fail, unless one of the categories represents "none of the above" and the classification method provides special support for this outcome [45]. And even in that case, bounding either the system's uncertainty that it has chosen the correct class out of the *n* available or the system's overall performance on the concrete answer "unknown class" is epistemically challenging due to the fundamental limitations described above in Section I.B.

III. A TAXONOMY OF ASSURANCE MODELS

Our goal in defining *safety* for AI systems is to define properties we want the systems to exhibit or not exhibit and then to convince a skeptical expert that these properties will hold or will not hold under an assumed set of operating conditions. The result of this analysis is *assurance* of the truth of the property, a belief by a skeptical expert that the property will hold under the desired conditions as intended. Experts must be able to see as much of the system's details as are necessary to achieve assurance. These skeptical experts could be part of the design or test teams, representatives of oversight entities such as review and certification bodies or higher command echelons, or expert stakeholders in an organization. We model the consumer of assurance arguments as knowledgeable but skeptical of the system's behaviors and performance. Assurance to a non-expert or an outsider follows indirectly from the position of the expert in the assurance argument plus the structure of the system. For example, airworthiness arguments are made to an aviation engineer audience but presented to the FAA under the DO-178 process (for software) or to a competent safety panel as part of the MIL-STD 882 systems safety process; end-user assurance derives from the correct operation of these processes rather than direct comprehension of the assurance arguments (that is, assurance arguments need not be directly understandable to pilots or weapons systems operators, so long as the acceptance process for assurance arguments is itself trustworthy).

Assurance is a structured form of belief communication: an *assurance case* states the desired properties in the form of claims, which may be broken into supporting sub-claims. Each claim or sub-claim is then backed up with technical argumentation. The arguments are supported by evidence derived from the system or underlying accepted theory or background world knowledge.

Assurance arguments and evidence can come from a variety of sources throughout a system's lifecycle. The simplest form of evidence is *experience* – if a system has been used before for a particular purpose successfully or if a component has been part of a system for a long time, this can serve as evidence that the use or component is safe or satisfies the desired assurance property. Care must be taken to avoid confusing lack of experienced problems with lack of any problems – a risk of experience-based assurance arguments is that rare phenomena may not be well-supported by a system, but experience does little to demonstrate this (especially when performance is aggregated). In such cases, operational experience may need to be augmented with fault testing that focuses on simulated conditions explicitly designed to evoke phenomena that could trigger rare but severe hazards. An example of this kind of augmentation is use of the Chaos Monkey tool [17], which is used to deliberately inject computer crashes to test the resiliency mechanisms in large cloud computations.

Another risk of experience-based arguments is that risks may not manifest in prior operations or that operations can drift slightly but successfully from the safe envelope, leading to an incorrect belief that the lack of problems in available experience reflects forward-looking safety or that small excursions from the safe performance region are acceptable on an occasional basis. This *normalization of deviance* can lead to situations where systems are regularly operated outside safe conditions on the basis of experience-driven assurance arguments [18].

For AI systems, their novelty and short deployment histories in tightly scoped situations can preclude experience-based arguments, but this will change as the technology matures and experience is gained in using it. However, AI systems are always limited by the fact that their behavior is implicitly or inductively defined based on finite experience and must *generalize* to handle new and unseen situations or combinations of input. As all systems will generalize imperfectly, there is some amount of *generalization error* (sometimes referred to as *irreducible error* or *epistemic error*) that results from the gap between the features the model measures and the variables that actually control the construct or target variable of interest. As noted above, decisions about how variables are measured and how target constructs are operationalized have an outsized influence on controlling epistemic error. Some approaches exist to placing upper bounds on the amount of epistemic error, described in the section on hazard analysis and mitigation, but this is an area where more research is certainly needed. Malik provides an excellent survey of methodological limitations in AI systems, focused specifically on machine learning approaches [19].

A useful practical example of this phenomenon comes from acute-care medicine: several machine learning systems have long been created to predict the onset of sepsis in ICU patients.[2] While these tools were initially lauded for their ability to improve detection and management of the condition [20], [21], they have all proved sensitive to problems with generalization and drift [22]–[24]: either the scores become less useful over time in the same clinical context (perhaps because the behaviors of doctors change as staff rotate out over months and years) or do not apply well across contexts (e.g., a score that works well in one hospital does not work as well in a different hospital). Investigations into what drives this loss of performance suggest a variety of causes: overdependence on specific features; differences in the way that care teams make and record decisions leading to novel gaps in the system's world model across time and space; and even variations in culture and the capacity for involved humans to "repair" the tool into their work practices while accounting for its deficiencies. A separate assurance difficulty is raised by this example: because the tool is an aid to professional judgement, its use does not depend on an approval or assurance process (because the discretion of involved professionals captures all behaviors of the tool). Yet we know that the tool changes that judgement and thus that unaided professional judgement is not comparable with the human-machine team.

Another common path to building assurance in a system is through structured *test and evaluation* at the time of system or prototype acquisition. Test and evaluation processes evaluate a system at a snapshot in time to determine capability. By necessity, testing can only establish properties which are *testable*, that is for which a test scenario and suitable performance measure can be established. Often, such properties and associated test scenarios are enumerated in a *test plan*. Test and evaluation, especially with a robust test

[2] Sepsis is a common, treatable, but dangerous condition that results in a large fraction of in-hospital mortality due to the subtle nature of accurate diagnosis.

plan against which system designers and developers can establish visibility into the desired metrics, can provide strong evidence that a system meets its functional specification. For software systems, test plans might specify testable metrics the software should reveal in a test state; some development methodologies are even "test-driven", meaning that tests are used as ad-hoc specification of program functionality.

Non-functional properties such as cybersecurity, information privacy, and many properties desired of AI systems are notoriously difficult to test because they are not directly observable – in the language of measurement introduced above, these properties are not easily operationalized into useful measurements. For this reason, test and evaluation methods are limited in an epistemic sense. This limitation is well-captured by the engineering aphorism that "tests reveal the presence of problems, but not their absence". This is especially true for software, for which the fundamental limitation of decidability implies that tests cannot establish the truth of a property even though they can provide counterexamples. Test and evaluation methods are generally applied in a post hoc manner – test plans can communicate to designers and developers what test scenarios will be considered, but the tests themselves operate on a real system or component. Limitations of test and evaluation are of particular concern for systems that are delivered in a finished configuration, ready for use. Acquisitions requirements should be carefully scoped to support whatever information is needed to build assurance arguments that cannot rest on functional testing.

For AI systems, testing often relies on aggregated performance analysis across a set of test vectors or known test cases. This can worsen the epistemic limitation of testing methods as the generalization error cannot be measured directly in this way. Even methods such as cross-validation on held-out data do not provide more than an optimistic estimate of this error [19]. Further, aggregation can mask poor performance on classes of operational scenarios and even reverse the direction of correlation between input and target variables. We discuss these risks and the extent to which they can be mitigated in our hazard and mitigation analysis.

Beyond testing, assurance can be gained through structured systemic evaluation of a system in operation. In the world of AI systems, such evaluation is often referred to as *algorithm auditing* or simply *auditing*. Audit-oriented methods suffer the same epistemic limitations as testing and may further suffer from lack of visibility into the system's structure. However, audits have the critical benefit that, unlike tests, the scope of review can be adaptive to findings. Further, external audit by a trusted oversight body can bridge the gap between assurance convincing to developers and assurance convincing to outside stakeholders [25].

If test and evaluation methods are applied only post hoc, are there methods that can provide assurance prospectively? Yes. Design-level information can feed into an assurance case as a source of claims, arguments, and evidence. For example, so-called "by design" methods which aim to establish properties within a system's design may provide automatic structuring of claims, arguments, and evidence. Encrypting data and maintaining custody of the key separate from the ciphertext, for example, can serve as an argument in support of a data security claim. Observe that this claim is not a functional, testable claim – it is quantified over all possible adversaries of a certain structure and power, subject to the assertion that the adversary does not access the encryption key. This sort of prospective, design-oriented argument can augment assurance cases to expand the scope of epistemically grounded claims available but does so at the cost of adopting explicit assumptions.

This is an acceptable tradeoff: if assumptions are clear, the system's safety in a particular operating scenario can be better assessed and assured overall, even if little can be said about what the system's behavior will be outside the assumed performance envelope.

This kind of conditional assurance is common in automated control systems. For example, if the interval between service requests is no less than a specified minimum inter-arrival time and there are no hardware failures, a software and system design with fixed computing resources can be shown to guarantee a response within a specified maximum delay even in the worst case. Note that without these assumptions, no guarantee is possible: if the rate at which service requests arrive is not bounded, it is possible to overwhelm the finite processing rate of any given implementation and cause some transactions to miss their deadlines. This is one of the reasons defenses against distributed denial of service attacks are so difficult.

The example also illustrates another difficulty: in closed-world conditions, such as in controlled laboratory or factory environments, conditional assurance cases can be quite useful and practical. However, in open-world conditions such as the contested environments expected for many applications of AI desired by DoD, systems must provide alternatives for what to do when adversaries deliberately create conditions that violate the performance envelope assumed in a conditional assurance case. This may involve monitoring the truth of the assumptions and triggering actions external to the AI system when violations are detected. This forms a concrete connection between assurance cases and the design of tactics, training, procedures, and backup systems that may include human actions and hardware safety interlocks. If the assumptions on which a conditional assurance case is based have been made explicit and can be operationalized, this approach makes it possible to have a runtime warning system that indicates when the people in a human-machine team should come to high alert based on observed evidence.

IV. HAZARD AND MITIGATION ANALYSIS

We explore the range of unintended behaviors of AI systems described in the literatures of research and practice to understand the scope of hazards which might reasonably be expected and foreseen when developing an AI system. For each class of hazard, we further summarize the state of knowledge about how it can be detected and avoided or managed and mitigated. Armed with knowledge of possible AI system harms and the foresight to identify and mitigate them, developers of AI systems can be better prepared to assess the safety properties and functionality of their own systems.

A. ISSUES OF GENERALIZATION AND MEASUREMENT

As indicated above, AI systems model the world in several ways at once and the fidelity of that model is critical to their functionality. All data-driven AI system behaviors are based on the idea that the future of the present will be like the future of the past. This is unlikely to be true for events of concern to DoD. Only the future of the past is represented in available data. Thus, data-driven systems will fail to account for changes in the world or new scenarios not previously experienced. Such systems do not take abrupt time-varying phenomena into account. Yet there are possible real-world state transitions likely to have large effects on patterns of events in the world. For example, if an adversary develops a new vehicle or weapons system, an image processing system used as part of intelligence collection or target identification and discrimination will not be able to recognize the new capability even as a new capability. This problem manifests in several ways.

1. Probabilistic Behavior and Optimistic Performance Analysis

Many practical AI systems, including many machine learning-based systems, are described not by a concrete mapping from inputs to outputs but rather by a distribution of outputs conditioned on a distribution of inputs (or sometimes just on a single input). The inherently probabilistic nature of such systems means that individual-case failures cannot be corrected by pointwise fixes, as might be done in a traditional software system. Instead, performance must be analyzed in the aggregate, as the rate or likelihood of correct output. Such analysis naturally manages problems of *aleatoric error*, in which noise at the level of individual data items causes those items to be misrepresented. By aggregating such items into a distribution and modeling the distribution, AI systems can perform well overall even when the noise present in a particular example is unknown or unclear.

However, aggregated performance analysis yields several concrete risks. First, the nature of the aggregation matters. Consider a problem where the population of inputs is made up of several distinct subpopulations (e.g., in the task of identifying aircraft parts, parts may originate from one of a handful of different platforms which are substantially different, such as airplanes and helicopters or jet-engine aircraft vs rotor-driven aircraft). Especially when one subgroup is much more prevalent than another (say the system sees mostly fixed-wing aircraft parts but only occasionally helicopter parts), high aggregate performance may mask low performance on subpopulations (in our example, the model could get outputs wrong for *all* the helicopter parts but aggregate performance might still be good because the number of failures is small relative to the total number of times the model is employed).

This arises commonly in systems which process images of people: systems may report high overall accuracy but in fact show very low accuracy when considered only on dark-skinned or female-presenting individuals (and even worse for individuals with both traits) [26]. Aggregated performance is *optimistic* in such cases, since the cost of errors for less common or rare inputs is averaged away by good performance for the most common case.

This kind of behavior is a concern in military contexts such as surveillance, where the system is seeking evidence of rare events such as sightings of periscopes: available data will contain many more examples of empty ocean scenes than scenes containing adversary periscopes, and large numbers of varied real data samples of this type are not possible to obtain in practice. Although simulations may be used to fill this gap, there is no way to ensure that the distribution of the simulated data points will be the same or even close to that of the real data to be encountered in future situations that matter – those observed during actual conflicts yet to happen.

Another case where aggregate performance causes real military risk relates to concept drift (see Section B.1): systems which perform well in a training or evaluation context risk underperformance on "rare" phenomena that may in fact be more common in the contexts where they will be used. Consider a targeting system tested on land in a temperate zone but used at sea in the tropics or near deserts: atmospheric phenomena such as dust clouds could be rare during evaluation but common in actual use.

Aggregated assessment of data with disparate subcategories can even lead to deceptive results. Aggregation effects can create spurious correlations between variables or even reverse the direction of a measurable effect. This phenomenon, known under the name *Simpson's Paradox*, arises quite commonly in real applications. For example, in evaluating the gender parity of graduate admissions to a large university, researchers discovered that the aggregate rates of admission for men and women showed that the school admitted women at a lower rate than men even though each graduate department had admitted a larger fraction of its women applicants [27]. In another case, the features that determined likelihood of voting for an extremist party in a democratic election showed reverse correlation between returns across the entire country vs. across its major regions [28].

Simpson's paradox is a manifestation of a larger problem in data-driven sense-making: the problem of *ecological inference*. In short, the behavior of an aggregate measure over an entire population does not necessarily predict the behavior of the same measure on a structured subpopulation or an individual – there is no guarantee that the rate at which some phenomenon happens in a population implies that an individual experiences that phenomenon at the same rate or with a probability dependent on the rate. Yet assuming this is so is core to machine learning and most data science approaches [19].

Given that the relationship between the population of interest in each analysis or prediction and the aggregation of strata within the overall background population has such an outsized effect on the performance of a model, one might hope that there was an optimal aggregation structure available and that it would be easy to discover empirically.

However, this is not true: there is no natural aggregation within a population which gives the best or most desirable answer. This is a consequence of the lack of any natural clustering function for arbitrary data sets [29].

Managing these risks requires having access to domain knowledge. With that knowledge, one can perform a *disaggregated performance and error analysis* focusing on performance and error rates for each subpopulation of interest. Since there is no natural clustering of population membership into reasonable abstract subgroups, disaggregating analysis requires a priori knowledge of which groups to interrogate. A disaggregated analysis can compare performance numbers and error rates across groups to understand if overall numbers are masking poor performance in some places. Knowing where performance suffers, in turn, can help AI system developers determine the best interventions to improve performance, such as targeted acquisition of data from a particular population. Data scientists often conjecture that exploratory data mining in the form of unsupervised learning to find the natural clusters in the data associated with subgroups within a category may help analysts to find the relevant domain knowledge, but given that clustering requires choices on the part of analysts and there is often no natural scale on which to base those choices, it is likely that appropriate stratification can only be determined using domain expertise.

Another hazard of data-driven analysis is identifying *causal* relationships between observations. Purely data-oriented analysis can only determine when variables are related, not why they are related or by what mechanism the relationship arises [30]. The relationship between identified correlations and outputs is particularly fraught in machine learning systems, which can construct complex *proxies* for the target of interest, synthetic variables which are highly correlated to the target in combination even if each is only weakly correlated.[3]

Proxies are problematic as they may yield strong correlation-based performance within the proxy data when there is no cause-effect relation in the real world to justify that performance. For example, several studies purport to identify criminal tendencies in individuals based on their facial features using machine learning. A skeptical reader might observe that such findings recall the scientifically discredited discipline of phrenology. Upon closer inspection, all such studies have demonstrated hallmarks of *overfitting* the data, learning unrelated features such as differences in facial pose, clothing, or image background to distinguish between the classes (often, the data about the "criminal" class are acquired using mugshots and the data about the "non-criminal" class are acquired in natural settings or from the Internet). Whether systematic differences in features between data classes or other subpopulations matter for the goal of the AI system again requires domain knowledge and careful accounting of measurement decisions (i.e., knowing what properties the model is meant to operationalize).

Correlation-based results can be further identified through testing of the validity and reliability of the relationship: in the criminality-from-faces example, a skeptic might ask whether the relationship holds over new data or over data where confounding factors

[3] Proxies can also arise naturally when information about the target or about some feature is encoded into other features. See Section IV.B on bias below.

such as presentation are controlled, for example in well-posed official portraits rather than mugshots. Malik provides exquisite detail on the risks of correlation-based reasoning [19].

As noted above, the fundamental assumption made when building AI systems using implicit or inductive specifications is that the patterns described by these specifications will hold in the future as they have in the past. This assumption is generally false, but making it is useful in certain cases. Like all assumptions, it must be undertaken advisedly and with awareness: an AI system is merely identifying and replicating patterns in an automated way. Those patterns break down regularly. Models are static, but the world changes (e.g., an adversary develops a new system with new characteristics; vehicles subject to predictive maintenance are suddenly deployed in new environments or on different operational tempos; the nature, frequency, or content of background communication or selected communication in an intelligence sorting application changes as real-world conditions change); this change is sometimes referred to as *concept drift* or *model drift*.

All safety analysis is familiar with the encompassing problem of *practical drift*, where without supervision the system breaks down and ceases to follow its own rules and logics over time [31]. As with practical drift, the solution to managing concept drift is supervision and in-line control. However, for AI systems this is complicated by the fact that it is rare to have access, especially ongoing automated access, to "correct" behaviors in the wild that could be used to compare against pattern-predicted behaviors. Typically, if data with correct behavior labels exist, they represent a snapshot in time or some fixed period of time, and all labeled information will be used in creating or assessing the system. Once a system is in use, it requires an ongoing source of pattern validation information since real-world patterns may change with time. This is a hidden facet of many commercial systems, which rely on large corps of human workers to annotate operational data for validation measures [32]. Expenses associated with this activity are an incentive to limiting or ending it, thereby increasing future system risks.

Related to concept drift, AI systems can cause their own gaps in representing the world through feedback. If the state of the world as measured by the system is affected by the system's behaviors, feedback should be suspected. For example, "predictive policing" systems identify areas of a city where police should be sent to patrol based on historical crime data. But crimes are reported by police. So if police are sent to areas where crime was reported and not to areas where crime is not reported, the distribution of crime reports will become even more lopsided [33].

Managing this risk is a form of optimization described by the *multi-armed bandit problem* – there is some value to following the model's output and exploiting the information, but also exploiting only this information leaves open the danger of practical drift. To solve this, it is necessary also to *explore* the space, sometimes making "suboptimal" decisions which have the side effect of producing new information uncontrolled by the existing model. Strategies for managing the explore/exploit dilemma are well studied in a variety of disciplines from statistics to psychology.

2. Epistemic Error and Generalization

By reproducing patterns, AI systems assume these patterns apply beyond the scenarios considered in generating them. This, too, presents risk. As noted above, patterns may be incorrect, falsely representing relationships in the world and yielding an *epistemic gap* between the model's representation and reality. But even if we stipulate those patterns are valid, there is no guarantee that they continue to apply beyond the cases where they have been validated. Patterns must be able both to *interpolate* (i.e., properly handle cases "between" known cases) and *extrapolate* (i.e., handle cases beyond the domain of known cases).

At a systemic level, problems of extrapolation can be limited by filtering the cases shown to a component or limiting operation of the system to a given operational envelope. But both can be problematic, especially due to the existence of so-called *adversarial examples*, samples which are close to inputs with known desired outputs but for which models will give different outputs [34]. Adversarial examples can be thought of as a risk in and of themselves, perturbations which may be chosen by an adversary to cause a system to misbehave or suffer failure on purpose [35]. But they can also be thought of as a consequence of the high dimensionality of representation in AI systems, a sort of inevitable curiosity of the system structure [36].

Regardless, adversarial examples demonstrate that even inputs "close" to those handled well by the system can cause problems and interpolation is not necessarily easier or less fraught than extrapolation. Both interpolation and extrapolation are captured by the notion of *robustness* in AI, which describes the ability of a system's patterns to generalize beyond cases considered in development and evaluation and to respond well to perturbation (either adversarial perturbation or simply generalization). By its definition, robustness is not directly observable. *Adversarial robustness* refers to the property that an AI system retains its robustness properties not merely against the random vagaries of the world but against targeted perturbation by an adversary [35]. This is a serious concern because all known defenses against adversarial examples have been shown to fail [37]. A recent advance in this area has improved the situation but there is still not an insurmountable defense [38].

AI systems tend to be quite fragile in practice, and failures of robustness are many. One study found that rebuilding standard benchmark datasets for image analysis using new images led state-of-the-art models to poor performance, suggesting that high-performing models were not capturing conceptual information about the data but rather "learning to the test", or overfitting in order to perform well on the benchmark [39]. Recent work has shown that overfitting and existence of adversarial examples are inherent in the way deep learning attains its high performance with respect to prediction accuracy [40]. Others have argued that benchmarks are a fundamentally flawed way to compare performance because of their epistemic limitations [41].

Bounding, or even assessing, generalization and other types of epistemic error remains an open challenge, although techniques such as data set augmentation, ensemble methods, and the aforementioned measurement-oriented validation practices can all help.

B. BIAS IN AI SYSTEMS

An oft-mentioned hazard in AI systems is the presence of *bias* [42]. Bias has a number of meanings and definitions depending on context [43]. As it was first used in the context of AI systems, bias referred to artificially limiting the space of models a system could learn, such as by controlling inputs to a known envelope as imagined in Section III.A [44].

Another simple definition of bias is well framed by the measurement view discussed above: bias is a systematic deviation in a model from reality. Systematic deviations can happen because of error in collecting data (e.g., by failing to sample adequately or collecting data in a way that selects for particular characteristics) or because of modeling choices (e.g., by choosing parameters or other model characteristics which cause or reinforce this systemic deviation).

Examples abound: the above-mentioned predictive policing example shows how measuring reported crime can lead to a biased picture of where actual crime happens [45]; a system for predicting recidivism of arrested persons used data about previously arrested people in that jurisdiction to create a risk score, resulting in the scores being higher on average for Black vs. white arrestees, a bias created jointly by the gap between measuring "arrests" and measuring "crimes committed" and the differential arrest rates for these groups in that jurisdiction [46], [47]; systems used to screen resumes for hiring can differentially and automatically reject minority race and gender candidates [48], [49].

As these examples show, another common use of the term "bias" in AI systems refers simply to socially problematic behaviors of AI systems. Because AI systems are specified indirectly, such problematic behaviors can occur accidentally unless system designers, developers, and controllers take active measures to avoid them. Further, because the nature of what is socially problematic will vary based on the stakeholder concerned [50], such measures must be based on broad-based review processes or comprehensive policy-making-style collaborative requirements gathering [51]. There are several good summaries and surveys of data and algorithmic bias issues, including Bonchi, Castillo, and Hajian [52]; Mulligan, et al. [50]; and Hardt, Barocas, and Narayanan [53].

As in the case of understanding optimism in AI performance assessments, bias problems can often be observed through stratified, disaggregated error analysis: where and how does the system make errors? Are aggregate performance statistics matched by performance in sensitive subgroups? (Risks of aggregated analysis were described above in Section IV.A.1.) Many tools exist to perform this sort of analysis, but techniques and procedures for deploying those tools are still nascent. Ethnographic work on AI system developers reveals requirements and concerns unmet by current technologies [54]–[56].

One major unmet need for bias management and mitigation is the gap between so-called "fair" AI techniques (e.g., fairness-aware data science/machine learning methods) and the actual goals of bias mitigation. As noted in our discussion of measurement issues, models are meant to operationalize well defined constructs, and anything which undermines a model's ability to represent and measure a construct is suspect for undermining the model's validity and reliability [10].

Thus, "de-biasing" data or building "fair" AI systems may in fact be undesirable, since it reorients the problem (which is often reflective of a real problem in the world where a bias exists) not as a truth under which goals must be pursued and ultimately achieved and instead creates a counterfactual imagined world where the bias is not present [57], [58]. It is not immediately obvious that optimal decisions in this counterfactual world correspond to optimal decisions in the real world.

Further, it is not clear that making component-level behavior in a system "unbiased" will yield overall system behavior which is safe in the same way (or even that system-level behavior will not be problematic). Instead, it is likely best to treat decision-making in contexts where system-level bias is important as it would be treated outside AI systems: as a decision which must be made despite incomplete information or other sub-optimal conditions. Instead, a goal should be to identify bias and surface that information to system designers, developers, and controllers to aid their decision-making. Controls on bias could be technical, operational, or policy-driven in nature. Much work in machine learning and elsewhere in AI focuses on the issue of how best to operationalize "fairness" or some other goal of "unbiased" decision-making, but this is a false goal – fairness is elusive and subject to stakeholder conflict, and even if we could operationalize it, the resulting definition might not yield desired safe system-level behavior [50].

Bias is not only a problem for socially sensitive problems – it arises in military problems as well. A likely apocryphal story surrounds the development of an imagined early AI system designed to distinguish Soviet and non-Soviet tanks. As the story is told, because of limitations in intelligence collection, the photographs of the Soviet tanks were mostly taken at night while the photographs of non-Soviet tanks were taken under more favorable lighting conditions. In this way, an AI system which simply learned to identify lighting cues could perform reasonably well (on laboratory data) on the task of "identifying Soviet vs. non-Soviet tanks".

Although the system imagined in this parable likely never existed, it is easy to imagine versions of the same issue leading to component-wise or system-level fragility due to bias: a model trained on vehicles from one region (say, Eastern Europe) will not perform as well in other parts of the world where vehicle makes and designs are distinct (say, Eastern Asia) and a model aiming to do well overall might perform better for the region where more data were available. If certain systems or classes of target are mostly photographed from overhead sensors, identification systems might do poorly in ground-based environments.

Similarly, a recent Defense Innovation Unit challenge, xView2, aimed to create and use satellite imagery data to categorize damage from natural disasters automatically [59]. Yet, while the challenge makes use of imagery from around the world, it is not obvious that performance identifying damage from hurricanes in the United States will transfer to assessing damage from sandstorms in the Middle East, typhoons in East Asia, or earthquakes anywhere. In each case, the structure of buildings and neighborhoods and the disaster's effects on them are quite variable. Without an argument to the contrary, it should be expected that the model performs well on the segment of data representing the largest share of the data, but might perform less well for other segments, despite data diversity. But in each case, high aggregate performance might mask poor outcomes for a subpopulation of operational interest.

Lastly, a famous example of *survivorship bias* concerns a problem of armoring RAF bombers during World War II; bombers needed additional shielding to survive anti-aircraft fire but could not be armored everywhere as the added weight would reduce their endurance and limit their mission effectiveness. To optimize the placement of added plating, a group of statisticians surveyed bombers after action to determine the location of anti-aircraft hits, finding strong clusters of damage points on an abstracted aircraft outline.

The study group realized that this data represented not a lopsided distribution of damage targeting, but a lopsided distribution of aircraft survival once hit. That is, the fact that few planes returned successfully from missions with damage to their engine housings indicates that the planes which were damaged at those points did not return. The key insight is to model anti-aircraft fire in this context as damaging planes essentially at random. With this insight, it becomes clear that the new armor should be added at the points where the minimum damage is visible in the data (as planes struck in the other points did not return, indicating the value of strengthening those locations).

Survivorship bias is an instance of the broader phenomenon of *treatment bias*, where data bias arises from interventions. Bias is best interpreted in light of a causal model of the underlying phenomenon, underscoring the value of domain expertise in model validation [60].

1. Concept Drift, Distributional Shift, and Domain Drift

Another critically important source of bias to understand is *shift* in characteristics of the real world over time. A model may perform well and even be robust to transposition from laboratory data to real-world scenarios at the time the model was created, but models are inherently static reflections of their development processes. Machine learning systems merely digest training data into best-fit functions, relating cases similar to those seen previously to an appropriate similar output such as a classification or a score. Yet the relationship between prior cases and desired outputs can and often must change: the shape and styling of vehicles evolves across model years and the changing popularity of vehicle types (and so a tool which identifies vehicles must be updated to reflect this change); the language used in online conversation changes almost daily (thus a tool which predicts how popular an online post will be must be constantly refreshed); world events affect the importance of facets of strategic and tactical decision-making (for example, the word “pandemic” was a term of art for specialists in 2019 but a household banality in 2020 and after – language data reflect word use, so models trained prior to the SARS-CoV-2 pandemic that began in 2019 will not reflect realistic usage in the years after). These problems are often described under the banner of *concept drift* [61], [62], the idea that as the underlying world changes, the world model of any given system will become outdated and require updating and adaptation. More specifically, changes in the underlying distribution of data driving a system’s world model are referred to as *distribution shift* (sometimes, *covariate shift*) and can be measured and potentially mitigated through traditional approaches to the explore/exploit tradeoff in the multi-armed bandit and Markov decision process models [63]. Distribution shift can also mask the appropriate choice of baseline distribution against which to compare: imagine measuring the mean-time-to-failure of a temperature-sensitive aircraft component before and after the warehouse in which it was stored experienced a temperature anomaly.

Clearly, the distribution of failures will change, and if the distribution after the anomaly is modeled using the data from before, the model will under-rate the likelihood of failure. And worse than the model suffering concept drift, analysis which can only see the post-anomaly distribution will not be able to know that the pre-anomaly distribution represents the failure distribution for new parts. This problem is common for environmental measurements in particular: measurements of the abundance of animal or plant species in an area speak only to the abundance at the time of the census and will not reflect changes in abundance from a decade or a century earlier. The constant change of reality challenges the notion of a "normal" level for many kinds of measures and complicates the question of validating a model (since the world against which the model must be validated is itself changing).

One common approach to providing needed adaptation is to make systems learn from their experience through *online learning*, which takes additional information available into account when performing optimization or making predictions [64]. However, making a model adaptable in the field adds a novel dimension of risk, namely that the model's self-updating capacity will reduce its performance on rare instances or instances seen many update epochs in the past, a phenomenon known as *catastrophic forgetting* [65], [66], a problem that occurs in human cognitive models of the world as well as computational tools. Approaches to mitigating this phenomenon include coalescing learned parameters at key points and protecting them against future update, for example by bounding the rate at which they can be changed by online learning processes.

In some cases, real world distribution shifts follow known seasonal patterns. For example, grocery stores have learned to stock more pumpkins before Halloween and more turkeys before Thanksgiving. This kind of time-dependence can be handled in machine learning systems by including the time of year of the measurement as one of the attributes of the data.

Unpredictable events that can change real-world characteristics are more challenging. Improved mitigations for this kind of distribution shift are needed, particularly for military applications in which surprise has value.

C. TEAMING/AUTOMATION BIAS AND INTERFACE ISSUES

It is often suggested that human involvement in decision-making can cure hazards or at least control them in automated or autonomous systems. Such a claim, however, is belied by literature on the human factors of automation. Here, we survey the major issues in human-machine teaming hazards.

Together, these problems are often described as *automation bias* [67]. We disaggregate them to discuss *automation overdependence*, including problems of inattention; *mode confusion*, including problems of interface design; and issues of *trust in automation*, including problems of when operators do and do not pay attention to recommendations of automated decision-support technologies or autonomous systems.

For all these issues and all problems of automation bias, we must view their dynamics in light of the primary human-systems interaction conundrum: the *paradox of automation* or the problem of "automation ironies" [15]. In short, when work in a system is automated away from human workers, the power of humans to create system behaviors and outputs is generally extended (if automation reduced system performance, it would likely not be implemented). This often means that fewer humans end up responsible for more system behavior and output.

Thus, humans become individually more critical to the system and more responsible for its behaviors and outputs while simultaneously less aware of the system's state, as compared to work done without using an automated system. As a result, humans are de-skilled regarding the system control actions they are now more critical in applying.

Consider as an example the containerization of shipping vs. the older break-bulk system of cargo loading. In containerized shipping, a relatively small number of humans (compared to the number of longshoreman needed to load/offload break-bulk cargo) have minimal interaction with each piece of a much larger pool of cargo moving through the system. Unwanted activities, such as smuggling, are now separated from operators by abstraction in the structure of the system (a cargo crane operator has little idea of what is in a container, beyond knowing where it should be lifted from and where it should be placed afterwards). Thus, the system has traded efficiency of one kind (the movement of tons of cargo per unit cost) for problems of a different kind (inability to survey/scan the cargo for undesired contents, spoilage, or damage).

Managing the automation paradox requires the establishment of system structures amenable to control interventions that avoid and mitigate undesired behaviors. Such *sociotechnical control structures* form the basis of modern safety assessment and program development [68].

1. Automation Complacency and Overdependence

Humans who supervise automated systems are likely to suffer cognitive fatigue in this work, not because the work is taxing, but because it is monotonous and holding attention presents a challenge.

For example, consider the problem of a security guard monitoring camera feeds. The guard no longer needs to patrol the area to surveil it, but inattention to rare-but-interesting events is likely to make detecting problems a challenge despite the guard's extended capacity to oversee their bailiwick. Instead, the guard is likely to develop a belief that the baseline view where no reportable events are happening is the current situation.

This *automation complacency* can also affect operators of vehicles using so-called "Advanced Driver Assistance Systems" (ADAS), who may lose capacity for supervising the vehicle as they give over authority to the vehicle to operate itself in ordinary conditions [69]. Many ADAS and other vehicle control automation systems hand off control to operators when ordinary conditions abate [70]. However, when operators have diminished situational awareness and control capacity due to complacency from not having been involved in normal operation, they are unprepared to receive this hand-off.

This issue is at the forefront of policy debates surrounding vehicle autonomy before regulators around the world. Automation complacency not only leads to diminished capacity due to inattention and lack of situational awareness, but also to longer-term de-skilling of operators, who do not have the opportunity to practice the tasks they are supervising and thus may take inappropriate actions when control is handed off.

A classical counter-argument to issues of automation complacency is that they can be solved by careful human-system interface design such that operators receive timely and appropriate feedback and can avoid complacency [71]. Others have argued that because automation-based interventions are primarily qualitative in effect, function-allocation-driven methods for separating human and machine duties must necessarily fail [72]. Instead, mitigation likely lies between these views: careful assessment of system structure and the allocation of functions coupled with appropriate interface design to support operators and effective communication between humans and machines can support the development of safe, effective systems. However, automation complacency undermines the baseline claim that human oversight is sufficient for system control.

2. Mode Confusion and Effective Interface Design

Another issue that causes mishaps in human-machine teams is confusion on behalf of human or machine components about the state and future actions of the other components, a problem referred to as *mode confusion*. Combined with the related problem of automation complacency, mode confusion is the primary driver of accidents in human-machine teams.

A well-studied example is the loss of Air France Flight 447 [73], a commercial Airbus A330 flight which crashed into the Atlantic just after crossing the equator during a transit from Rio de Janeiro to Paris. A survivable failure of the air speed sensor led the aircraft's autopilot to disengage and transferred the fly-by-wire system from "normal flight law", in which the angle of attack is limited by software control, to "alternate flight law", in which the angle of attack is unrestricted.

The pilot, though trained and experienced, spent several minutes attempting to "power out" of a stall that resulted from throttle slowing prior to autopilot shutdown (by pulling up and throttling up). Though an appropriate maneuver for the A330 at a limited angle-of-attack, in alternate flight law (and at night over ocean, when attitude indications are limited to instruments) the pilot took the angle of attack higher than the plane's thrust capacity could handle to regain airspeed and stop losing altitude. Worse, a problem in the interpretation of extreme values led the aircraft's "stall" indication to turn off when the plane was nosed exceptionally high, causing a situation where the falling aircraft would appear to enter a stall when the angle of attack was reduced. Although cockpit voice recorder data indicate that the pilot not flying read the mode transition off the aircraft's cockpit console correctly and that this transition was acknowledged properly by the pilot flying, confusion over how to respond to the scenario is clear in the minutes between the autopilot disengagement and ultimate crash into the ocean.

A similar set of problems occurred in the at-sea collision of the *USS John S. McCain* (DDG 56) with the *M/V Alnic MC*, a chemical tanker, in the heavily trafficked area near the Strait of Malacca. The *McCain*'s bridge crew adjusted their throttle to handle contact with the *Alnic MC* but did not realize that their integrated bridge and navigation system (IBNS) was set so that the *McCain*'s twin screws were not "ganged" (i.e., locked to the same throttle setting). Thus, while intending to reduce speed for a passing maneuver, the bridge crew had inadvertently put the ship into a turn.

Subsequent efforts to recover the situation failed due to confusion about whether the active control console was the main helm or the lee helm. As a result, the *McCain* swung athwart the *Alnic MC* and was struck alongside, resulting in 10 fatalities and an estimated $223 million in damage to the ship.

Managing mode confusion requires similar attention to human-system interface design, appropriate feedback without engendering complacency, and human-factors analysis of the system as fully constituted to determine training and doctrinal requirements. Unlike automation complacency, mode confusion can be lessened by training and experience.

3. Trust in Automation

Beyond managing the confusing aspects of the human-system interface, human-machine teaming can be viewed through the (relatively) simpler lens of trust: do humans believe the machine components of a system will behave as expected and are they willing to rely on this belief? Trust is a psychological property of the involved human or of outside human stakeholders (e.g., lack of discrimination in a criminal justice risk management score is relevant not only to people in the criminal justice system but also to ordinary citizens who may be concerned about equality of opportunity). As such, trust is not a property that can be built into a system or even necessarily assessed and validated through assurance processes.

Instead, assurance processes can only assess the extent to which a system's design is supportive of trust or creates the conditions for formulating trust, a system capability referred to as *trustworthiness*. Most simply, assurance can identify cases where trust would be undermined and flag them for remediation.

An important hazard for any new technology is that it may not be adopted even if adoption could improve progress towards the potential adopter's goals. Trust is generally a prerequisite for adoption – if stakeholders do not believe that a system will advance their goals, they will seek alternative means. AI systems can fail to engender trust by lacking assurance, by performing poorly, or by seeming to perform poorly by mishandling common cases or overwhelming humans with false alarms.

Alarm fatigue, a condition in which human operators ignore guidance from automation because they consider the guidance low quality, is a common problem for automated systems. Operators in a security operations center (SOC) or network operations center (NOC) are quite familiar with the volume of alerts and information that must be processed to perform their jobs. Many of these alerts represent minimal risk events but must be reviewed anyway because they may represent higher risk events under certain circumstances not captured by the automation. If systems raise alarms repeatedly and operators determine repeatedly that the alarms are false alarms, operators are likely to start ignoring the alarm.

Famously, in construction and mining settings, the alarms that sound when vehicles are backing are so common they get treated by workers as background noise. In 2009, a train on the Washington, DC metro system collided with a stopped train despite the presence of automated train control capable of detecting trains on the same track segment and forcing them to stop. The accident caused 9 fatalities and over 80 injury casualties. Prior to the accident, an alarm was sounded to train dispatchers that a faulty track circuit existed at the location of the accident and a stopped train would not be detected. But dispatchers were overwhelmed by such alarms, which happened at the rate of approximately 8,000 per week and were not treated as unusual [74].

Managing alarm fatigue requires assessing the tradeoffs between false-positive and false-negative alarms and their consequences and possibly adjusting alarm thresholds to tune how often operators observe such alarm failures. Accuracy is a key performance parameter in this context, since reducing erroneous alarms reduces the opportunity for fatigue. Assessment of the effects of false alarm rates on human workers is a key factor in risk mitigation decisions for this type of hazard.

All automated systems, including AI systems, operate according to pre-determined rules that, at the most flexible, map inputs into distributions over outputs. These rules are fixed into the structure of the system and its components and cannot adapt as context changes. Thus, machines are often faulted for operating without concern for context.

Trust, therefore, requires that a system be capable of incorporating context on an as-needed basis [75]. Context can be incorporated by enabling human overrides of system decisions either at the moment of decision or after. Managing context also requires that mechanisms be present where stakeholders in the system can identify situations where context was insufficiently well handled and the system's controllers can act based on this. When the effects of system behavior are not reversible (such as when the system controls the use of lethal force), post-hoc rectification of mistakes is not possible, so pre-action confidence must be correspondingly higher to enable trust.

D. AI SYSTEM SECURITY AND SUPPLY CHAIN SECURITY

Increasing concern surrounds security problems in AI systems, ranging from the manipulation of the data representing and controlling the system's world-model to manipulation of the system's components themselves to manipulating system inputs [76].

Another class of risks not captured with immediacy by this taxonomy is risk to AI system supply chains and dependencies. Much of that risk is inherited from the software nature of AI systems and mitigations fall along the same lines as mitigations for software supply chain security risks.

Here, we consider risks that are specific to AI systems, especially to data-driven ones as machine learning is often the technology of interest for adopters of AI and because data control the world-model of even model-driven system architectures. AI systems, and especially data-driven AI systems, carry a number of information security and privacy risks [77].

1. Data Poisoning and Manipulation

As AI systems are specified and evaluated using indirect framing of a world-model, they rely more heavily on the integrity of data representing that indirect framing. If data are modified in such a way that it changes the world-model of the resulting AI system, this is referred to as *data poisoning*.

Data poisoning is an example of a violation of data *integrity*, an important element of information security recognized classically. However, while good tools exist to manage data integrity across time and space, those tools are not used as widely or as effectively as tools for data confidentiality.

Unlike data breaches, in which the release of confidential data can be seen and identified, it is often difficult to obtain evidence for attacks on data integrity. The extent to which cybersecurity compromises of AI systems or their supply chains leads to attacks on integrity is largely unknown.

The simplest approach to managing data poisoning risks is to control data integrity across the lifecycle of data. However, an absolutist approach such as this is not necessarily consistent with the flexible decision-making goals of AI applications or the risk-based approach to hazard management commonly used for safety program development and administration. Thus, the machine learning community has developed measures for determining how effectively an adversary can poison data with the goal of corrupting the behavior of resulting models, allowing the risk of system behavior corruption to be made amenable to assurance argumentation [78].

However, a complete assessment of data integrity risk is likely advisable regardless of such mitigations, as many assumptions underlie the bounds provided and those assumptions require justification.

2. Model Manipulation and Extraction

Integrity attacks can be performed on models as well, although direct modification of model source code and parameters is generally considered under the bailiwick of AI system supply chain security. However, the risk posed by models to the disclosure of underlying data is a separate hazard.

Direct disclosure of data obviously reveals information contained in the data, but models depend on the data and thus are informationally related to it. Thus, even giving an adversary the opportunity to query a model with partial data can be enough to allow inference of other possibly sensitive data in other inputs or in the model's training data. Research shows that such leakage is possible in practice, and in targeted ways [79].

The proposition that models contain information is not on its own surprising – if they did not, their performance at predictive tasks would be no better than random. What is interesting is that this information can be extracted efficiently. This problem impacts not only AI systems but even systems designed to protect the security of the underlying data, such as encrypted databases that still allow queries to be constructed over the encrypted data [80], and the efficiency of extraction attacks is quite high.

This is significant for Navy applications because extracted knowledge of the models used by deep learning applications can be used to construct adversarial examples that can be used to attack the AI systems. It may also mean that the ability to interact with an AI system is sufficient to reveal information about sensitive data used to construct it.

3. AI-Specific Supply-Chain Security Issues

Beyond the ordinary supply chain risks of any software [81], AI systems have particularly extensive and brittle supply chains and care must be taken to establish the sources of risk attendant to this fact. AI development often uses a large footprint of library code which is under-evaluated for cybersecurity risk; many data processing and AI development infrastructure tools lack basic security capabilities such as the ability to authenticate clients or control access to portions of the data. In cases where these capabilities are added over and above the system's design, care must be taken to determine whether the composition functions at a system level in all and only the intended ways. It is likely that novel library code should only be introduced carefully into secure computing environments, which limits the capability available in those environments. This is an area of active research and it is under-attended by the community. Managing the risk/benefit tradeoff of additional functionality vs. marginal security risk represents a challenge in any information system. Unlike system safety frameworks, tools for assessing cybersecurity risks (such as the RMF) are not known to improve outcomes or reduce system hazards.

4. Privacy as an AI System Hazard

AI systems, by their nature, identify and extract patterns in the world to develop new and better approaches to achieving their systemic goals. As a result, much has been written about privacy in automated systems, especially AI systems, over decades [82].

However, privacy as constituted in the literature is generally about the informational autonomy and control of *individuals*. From a DoD perspective, individual privacy may be beneficial because of compliance requirements or as a structure for supporting trust in programs and processes. But often, DoD's operational concerns require privacy where benefits accrue to individuals only by dint of their relationship with an organization.

Information in patterns identifiable in data can reveal facts such as changes in operational tempo (which might reveal sensitive operational security information such as impending operations), organizational charts and hierarchies, or even reveal classified facts that are inferable at high confidence from data that seem non-sensitive.

Frameworks for evaluating and managing privacy risk, such as the NIST privacy framework, only barely consider these risks and only when evaluators read the risks into the framework, making use of the framework's flexibility to add concerns rather than shed them [83]. More emphasis on these issues together with supporting policy may be advisable.

E. FORMAL VERIFICATION OF AI SYSTEMS

Although the non-declarative nature of AI system specification makes traditional specification-oriented formal verification methods difficult to apply, that does not mean that some progress has not been made. Indeed, several different lines of effort show interesting and useful results even at the scale of usable realistic systems. We discuss approaches made to-date, contextualizing them in the larger system safety problem as we have framed it.

1. Verifying Model-Driven Architectures

a. Verification for Expert Systems

A classical form of AI is the so-called *expert system*, which uses a database of facts combined with rules of inference to build new conclusions automatically. Because the developer of an expert system has control over the set of facts to include and the decision rules by which the system can extract behaviors and recommendations, it is possible to ask questions about the detection of anomalous outputs or the completeness of the output set according to various metrics [84]. In specific, validation and the issue of model gap management remains paramount for these systems, as any metric of completeness must carry assumptions about the proper modalities for representing the world. Although expert systems represent the world symbolically, interpreting their recommendations can prove difficult – a decision reduced to a large decision tree can clearly be traced through the tree node-by-node, but such a tracing does not explain why the tree has its particular structure and only provides instrumental contrastive reasons for why a decision is not different, located elsewhere in the tree [85].

Expert systems are often used for decision support, to replace or augment human expertise. An important case is diagnosis and triage during vehicle maintenance – a system might diagnose a problem based on symptoms or recommend the next investigative/diagnostic procedure to run. If a system is incorrect, not only could that harm readiness by extending service periods but it could increase the cost of maintaining vehicles, so verification and validation are both of critical importance (i.e., the system must both accurately represent vehicle failures and diagnose interventions at the correct point in a vehicle's maintenance and lifecycle history). Expert systems also show the limit of focusing on verification and validation (whether in a formal sense or not) in terms of evaluating a system for higher-order properties such as credibility to users and other stakeholders (e.g., a maintenance diagnosis system must not only garner the trust and use of mechanics and maintenance technicians but also vehicle operators, program sponsors, and ultimately commanders who are responsible for unit readiness). Such higher-level and whole-system assessments depend on V&V, of course, in the sense that V&V failures will undermine higher level judgements. But formal verification on its own does not furnish credibility or positive evaluations of things like value for sponsors [86].

In our taxonomy, expert systems use a *model-oriented architecture* despite relying heavily on the contents of the knowledge store for their actual behavior and function. For this reason, as noted above, it is straightforward to reason about failures since specific behaviors of an expert system are straightforwardly traced, but it can remain challenging to establish whether the knowledge base itself is prospectively correct even when it is the focus of verification and validation efforts [87]. In this way, we see that the distinction between data-oriented and model-oriented architectures is not necessarily clean (model-oriented architectures are often at least post-hoc interpretable, however). Instead, verification of the knowledge base must proceed from external knowledge of the structure of the domain. It is from this structure that the soundness and completeness of the facts in the knowledge base can be attested, measured, and confirmed. Without such domain assumptions, there is little meaning to claims of verification or validation. Thus, even expert systems and model-driven AI tools must carry assumptions about their domain of operation to be assessed for safety. Beyond soundness and completeness of the data with respect to real-world knowledge, knowledge in an expert system can be *redundant* (representing the same fact about the world in multiple imprecise ways when one precise statement would be preferable) or *inconsistent* (representing a fact about the world in two incompatible ways, such as labeling distinct items with the same item in a knowledge ontology or labeling the same item in distinct ways). All these can lead to problems with the correctness of the system's behavior.

Having established both the domain dependence and the modalities of failure, it becomes possible to gather "meta-knowledge" against which to verify and validate the knowledge in the expert system. Domain-independent V&V is also a known approach, based on heuristic detection of *anomalies*, or unusual uses of the knowledge schema. Anomalies are only potential errors – they may be intended behaviors (this recalls the concept of the "normal accident" or "system accident", a rare behavior allowed by a system's design which is harmful, unintended, and the result of insufficient safety control [88]). Anomaly detection often relies on simple checks, such as for consistency, completeness, and correctness of the rules in the knowledge base. For example, a correctness check might examine generated statements for the lack of *circular reasoning*, or chains of rules which feed back to the same knowledge state. Although the techniques identified above can be helpful, they do not constitute a complete verification framework, and do not provide absolute guarantees of safety.

2. Verification in Reinforcement Learning

In cyber-physical systems, a successful approach to controller learning comes from *reinforcement learning*, a technique in which the controller maximizes its total *reward* when each action is assigned a value under a *reward function*. Reinforcement learning systems tend to perform well in unconstrained, unmodeled, open-world environments, but because of their unconstrained nature do not provide any meaningful guarantees of safety or even bounded control [89].

To sidestep this problem, work on verification in reinforcement learning often uses *runtime monitoring* so that the controller can be treated as having a *nondeterministic policy* with multiple safe actions. Verification, rather than showing that the optimal control policy is safe, merely confirms the set of safe actions at a particular state and confirms that the runtime monitor will intercept the system's action if it is not deemed safe (thereby allowing normal optimization within the reinforcement learning process to solve the optimization problem). This is the paradigm of "Justified Speculative Control" which pairs a formally verified controller and an ordinary reinforcement learning algorithm in a sandbox in such a way that proofs about the verified controller transfer to the optimized reinforcement learning model [89]. This technique forms the basis for application of a wide variety of formal methods and shows promise in safety analysis and verification for even real-scale cyber-physical systems [90].

As with expert systems, while careful architecture and proof construction techniques can make systems amenable to verification, validation for these systems remains an important and difficult problem, due to the fundamental model gap issues identified in Section II and especially Section II.C.

3. Verifying Data-Driven Architectures and Machine Learning Models

The model gap issues of model-driven architectures are substantial, although detected failures at least correspond to interpretable behaviors of the model. By contrast, machine learning models use a data-driven architecture: instead of specifying the contours of the model and leaving the implicit portion of the behavior to the operation of the model, data-driven architectures specify only the data and the processing methods to be used (typically, optimization of some objective function). Here, failures of correctness may not be obviously attributable to these choices, and failures are often attributed to "bad data" – data that are incomplete, biased, or otherwise mismeasure the world and reify some kind of modeling gap (data are better described as *made* rather than being *found* – although it is comforting to think of data as an objective reflection of the world, they represent a model already and that model may bring in/leave out critical world knowledge [91]). However, the verification situation is not entirely bleak: machine learning models can often be verified to perform within stated performance envelopes. We describe the major approaches shown in the literature.

a. Interval-Bound Propagation and Robustness Guarantees

An important class of robustness failure in machine learning models is the *failure to generalize* appropriately. That is, models perform well on data similar to that encountered during training, but because they are limited to a fixed menu of output behaviors will choose the maximum likelihood output even when this likelihood is very low because the input is unlike any training examples.

To avoid this type of failure, it would be beneficial from a verification perspective to define a performance envelope in which this sort of failure will not happen because data encountered match the training data well (this approach is very natural – many systems are certifiably safe only based on assumptions about when and how they will be used). Such an envelope could be established through input filtering – if inputs (do not) match filter predicates, they should be discarded or the machine learning output marked as possibly erroneous or even treated as an error. However, the problem of *adversarial examples* defeats this simple solution. Adversarial examples are inputs which are “close” to training inputs as quantified by distance measures (such that they might count as the same class to a human) but on which the model reports a different output [92]. For such examples, filtering is insufficient for defining a performance envelope (since models, being susceptible to adversarial examples, do not learn the filtering function). Adversarial examples are often presented as a kind of “attack” on systems rather than inputs with the property that they are misclassified despite being norm-wise close to known inputs, suggesting that they are less exploitable bugs in the system and more system behaviors that must be managed [36]. Managing them requires overcoming the problem of oversimplistic definitions of safety envelopes.

One technique for establishing such a robust safety envelope for machine learning models is “interval bound propagation (IBP)” [93]. In specific, IBP provides a guarantee that a model’s output will be stable relative to norm-bounded perturbations. IBP allows a modeler to efficiently build models that are optimized with regard to adversarial perturbations. It is not a *guarantee* that the model is without perturbations within the specified norm-ball of known examples, but unlike complete methods (described below), it does scale to large model sizes. However, problems with model robustness are likely the result of problems in validation and the inclusion of non-robust features [36]. Indeed, sensitive dependence on incidental features drives performance issues that undermine even the reproducibility of machine learning results [39]. Other work has investigated the use of ideas from robust optimization to manage this issue [94]. However, recent work suggests that adversarial examples may be an inherent feature of deep learning [40]. We recommend further investigation of the effectiveness of these techniques in contexts relevant to DoD before they are incorporated into safety assurance guidelines and procedures for military AI systems.

b. Explicit Model Checking: MIP and SMT Approaches

Although explicit specification for the input-output relation of a machine learning model such as a neural network is not possible *a priori*, the model nonetheless captures some function and thus it is possible to make claims about that function, such as that a certain class of input must be mapped onto a certain class of output [95]. Armed with an imputed specification which *models* a function, one can attempt to do explicit *model checking* to verify that a machine learning model matches the modeled specification in a complete and sound way. Progress in this direction has happened through the use of mixed-integer programs (MIPs) [96], [97] and satisfiability modulo theory solvers (SMT solvers) [98]–[100]. An advantage of the model checking approach is that it offers complete verification of a claim, drawn from a general set of possible claims, over all possible inputs to a machine learning model. The main disadvantage is that this completeness limits the scalability of the verification approach. However, work in this area has been able to demonstrate verification of properties with some interest, such as the control of a flight collision avoidance system [98]; that system is still small by the standards of modern machine learning tools, but its behavior is rich enough that it captures a real-world problem well. Thus, scaling the model checking verification approach could be achieved by pursuing simpler models that are similarly performant [101] or by finding refinements and approximations that retain soundness for model-checked safety properties [99], [100], [102], [103].

Many recent surveys document approaches in the domain of verification of machine learning and neural network systems [95], [104]–[107]. Other work applies ideas from formal verification to control problems in reinforcement learning [89], [90].

F. INTERPRETABILITY AND EXPLAINABILITY

Much concern has been raised about the safety issues arising from the so-called “black-box” nature of AI systems [108]. In short, the purported concern is that because AI system behavior is complex or not visible to those who must interact with or are affected by the system, it is difficult to make sense of the system’s behaviors [109]. This leads to two safety concerns: first, that it will be difficult to know when a behavior is not justifiable or might represent or lead to an unsafe state; and second that systems that are misbehaving will be difficult to debug or override during operations.

As noted in Section IV.C, there are substantial issues in composing automated systems and human operators. Questions about interpretability and explainability are related but distinct – unlike the issues in Section IV.C, issues of interpretability and explainability have to do with the psychology of the operators rather than the structure of the system. Failures can occur either in the structural ways that humans and machines relate to each other (not affording humans sufficient situational awareness to take over the disengagement of automated control in a vehicle, for example) or in the perceptions of the operators (an operator misinterpreting why a particular decision recommendation was given and therefore acting incorrectly as a result) [110].

Beyond this, AI systems can and do acquire unsafe behaviors because of inaccuracies in their inductive specifications, and difficulty in explaining or interpreting system behaviors may present risks that manifest at the development stage. To take one well known example, although a neural network provided the most accurate decision recommendations in a project to use AI to support emergency room medical clinical decision-making, doctors rejected the system on the basis of not being able to understand the rationale behind the recommendations [111]. Specifically, the project aimed to recommend to doctors whether to admit a patient from the emergency department as a hospital patient given their pneumonia diagnosis, symptoms and medical history. Some pneumonia patients will recover safely at home given a course of antibiotics and bed rest. Others may fall into a crisis within a day and require critical care intervention to avoid severe risk of death. Experienced doctors have difficulty distinguishing between these classes of patient, and a decision aid was sought to improve sorting outcomes.

An assessment of the proffered decision-support models revealed that a rules-based model predicted that the clinical feature of having asthma predicted low risk from the pneumonia diagnosis. Interdisciplinary consultation confirmed that this learned relationship is medically incorrect: asthma patients are at higher risk of complications from pneumonia. However, asthma patients as a group did have better outcomes in the data on which the AI systems had been developed. Why? It turned out that the hospitals from which the data were gathered had policies requiring the immediate admission of such patients regardless of other factors (in some hospitals, these patients were put directly into critical care). Because of the stronger intervention, this cohort suffered fewer complications overall. But because the AI systems could not observe this intervention in the data, they developed a large model gap that could not be detected by analyzing the model or the data alone. Subsequent research demonstrated tools which could provide an AI system with similar performance to the neural network but with greater capacity for interrogation and debugging [111].

This example shows a dichotomy in the value of explainability for contributing to trustworthiness in an AI system: should explanations be given to end-users, who might then use a kind of "common sense" to understand when the machine components of a system should be overridden by human judgement, or are explanations better suited as debugging tools that help system developers and controllers understand why failures have occurred. Observe that in the first of these two contexts, we encounter the human factors problems described in Section IV.C [112], [113]. And in the second, what development or operations staff require to understand failures is better captured in the broader concept of *traceability* within the assurance process [7], [114].

However, the broad concept of explaining the behaviors of AI systems as part of understanding their trust and assurance properties remains important, both because it helps define a wide range of trust requirements for such systems [85] and because the large number of techniques developed under this research program show promise in improving AI system reliability and provide levers to dislodge the difficult problem of model validation [115] (as noted above and below, although verification of AI systems is difficult, it has proved more amenable to progress than validation in existing research and practice).

It is also possible to view the distinction between the requirements for human-level engagement with an AI system and the value of a mechanistic description of the input-to-output relation and workflow as a dual set of requirements which can be valuable simultaneously. Psychology teaches us that both sets of requirements are important to building the human-system interface [110], the first because humans must contextualize the AI system's behavior in order to react appropriately and the second because some humans require the detail of a full explanation in order to themselves form the needed interpretations.

Work in this area remains important and under-developed, especially as it relates to assessing human-machine teams and within that especially as it relates to decision-making and decision-support systems with automated components. In both the psychology and computing literatures, there is much confusion around uses of the terms "explainability" and "interpretability", with many works conflating these terms in practice while simultaneously identifying them as distinct ideas [116]–[118]. Research on applying these ideas has pushed beyond such abstractions toward applications, particularly in the medical domain [119], and toward ideas about managing the generalization and common-sense application processes within the machine components of the system [120].

G. ROBUSTNESS, RESILIENCE, AND PERTURBATION DEFENSE

A major open problem in all manner of AI systems is the problem of *robustness*, the problem of knowing whether the performance of a system will remain stable over various sorts of generalization (e.g., generalization to new data beyond what is used in development and evaluation; generalization to new-but-related domains such as new populations with similar features undergirded by related constructs; stability of performance over time as relationships operationalizing the construct of interest evolve). That is, robustness is the property that an AI system will remain performant even when it is *perturbed* – given unusual input or applied in situations other than those it was designed for.

Robustness is a key property for safety: hazardous system states likely do not resemble the states for which a system's performance has been evaluated (this is almost definitional – in expected states, the system's performance has likely been assessed correct, while hazardous states where the system may misbehave are states that were unexpected even if they are allowable as states of the system [88]). Although there is no general technique for establishing the level of robustness afforded by a particular AI tool nor a universal defense for sustaining performance against the most destabilizing perturbations, much work has shown how to understand the envelope within which a model will be stable, how to improve the level of robustness an AI system provides, or how to detect that a tool is being asked to perform outside its safe envelope. Many research problems remain in this important and underattended area, however, and while we aim to taxonomize these questions and what is known about their solutions, we can at most scratch the surface of a deep, unsolved problem that represents a classical challenge for all sorts of modeling. The robustness problem's difficulty underscores our focus in this report on explicitly examining a system's assumptions and its world model to understand the applications where it may be employed safely. The opposite of robustness is *fragility* or brittleness; many AI systems are inherently brittle, and understanding in what ways they might catastrophically stop working is key to determining when AI systems can be employed usefully and reliably [4].

A system-level concept which is analogous to robustness is *resilience*, which can be defined in many domain-specific ways but which in general refers to a measure of a system’s ability to survive and persist in driving towards its goals even in a variable environment. Resilience is little-studied in the field of artificial intelligence, but systems which manage safety-critical applications must be analyzed for resiliency at the system level (which may be driven or affected by robustness at the component level – certainly, non-robust components can cause failures of system-level resiliency, but the converse may not be true). Resilient systems fail gracefully and gradually [121]. Systems generally require design and active management to support resilience. Resilience is not the same as static stability or lack of response to perturbation; rather, it is the system’s ability to recover normal operation after a perturbation was encountered. The study of how to measure and understand resilience is underdeveloped, both for AI systems and in general.

Much of what is known about robustness comes from the study of *adversarial robustness*, or robustness against perturbations specifically designed to maximize the misbehavior of an AI system [34]–[36], [92], [94]. The study of so-called “adversarial examples”, or inputs from a valid domain which trigger adverse robustness behavior despite being similar to inputs on which systems are performant echoes a longer history of work on “evasion” of AI systems [122], such as by spam in communications systems, malware in computer networks, and camouflaged targets or targets deploying countermeasures to detection in military applications. Each of these cases is well studied, but study and practice in real-world applications have yielded neither complete defense nor fully general systems which perform well even when adversaries are allowed to choose inputs. Even in the space of adversarial perturbation defense for advanced technologies enabling the latest generation of AI systems, such as deep neural network machine learning, an enormous amount of study yields neither complete defense nor strong robustness [38].

What is notable about the field of adversarial example defense for machine learning systems is that the focus of the efforts to resolve it are focused primarily on the problems of generalizing defenses or on the problem of improving model generalization. Yet these framings are the most difficult epistemic approaches, substantially more difficult than attacking the problem at a systemic level. Consider evasion attacks known in earlier systems, such as the injection of spam into content networks and communications channels. It is well known that the deployment of anti-spam measures increases the volume of spam sent while modifying the distribution of spam approaches in order to maximize the adversary’s chance of evading the defense [123]. Thus, efforts to harden communications systems against spamming must account for the ways new defenses will reshape the nature of content injection. The same goes for AI systems: defense against adversarial inputs may be as (or more!) brittle to changes in attacker behavior as the initial system was to misbehavior from attacker-crafted inputs. And just as the solution to spam problems often lies in making the spamming activities unprofitable, there are likely system-level equilibria that are robust and resilient to adversarial actions. Yet such design-level approaches are rarely studied. We leave the vast task of summarizing what is known about specific adversarial attack and defense to another technical report also stemming from this project [38].

Related to assessing robustness and the risk of generalization error is the problem of bounding how much an AI system’s performance will be affected by generalization or perturbation of any kind. We stress that our core argument – that all AI systems rely at least implicitly on a model of the world which should be well aligned to underlying constructs and assumptions – is the core mechanism by which it can be hypothesized that an AI system will generalize beyond the training data effectively.

Further, understanding the assumptions and the necessary modeling gaps created by a system's world model points the way to reasons a system may not generalize well. For example, a system which identifies the make and model of road vehicles may generalize poorly if applied in a different country than the one in which it was developed as vehicles in the new country may look very different (or the same vehicles may be badged as different models in the second country). But it may also perform well if the second country purchases many of its new and used vehicles from the same manufacturers as the first country. Although there is a substantial amount of research on measuring uncertainty in modeling of all kinds (including machine learning), the area does not provide a complete answer to the core question of model validation [124]. Even advanced Bayesian methods of uncertainty estimation cannot give a full answer, partly because their own assumptions must be validated and partly because viewing the problem at the component level does not provide the needed context to understand the way the system as a whole operationalizes a world-model [125]. Systems-theoretic validation approaches hold promise for recontextualizing component-level requirements in light of system-level paths to hazardous states [68].

Another approach to improving robustness holds that, since causal relationships are likely to hold up when the context of a system's application changes or when practical drift modifies or invalidates system assumptions while non-causal relationships are likely to break down, the appropriate point at which to separate robust from non-robust models is the distinction between causal relationships and incidentally discovered, non-causal relationships such as incidental correlations. The question then becomes whether causality in the structure of an AI system's world model can be inferred automatically or whether the concepts that underlie causality must be taken on as assumptions. There is an enormous literature on the automated inference of causal relationships, drawing on a long history in science and philosophy [126]. In specific, Pearl's calculus of causal reasoning [30] has launched an entire methodological subfield in machine learning [127]. Other methods from statistically controlled natural experiment theory can also be applied, such as the theory of instrumental variable analysis [128]. This line of research has led to an enormous field of counterfactual causal reasoning to identify failure modes and improve predictive fidelity [129]. However, because counterfactual reasoning departs from the operationalization of an AI system's core world model, it has only limited validity in many cases [130].

H. AUDITS, ADVERSARIAL EVALUATION, AND DOCUMENTATION

An often-suggested class of intervention for risk management in AI systems is to *audit* the systems, either collaboratively [131], [132] or adversarially [133]–[137] (settings which roughly correspond to "white-box" auditing/testing, where tests can depend on system internals, and "black-box" auditing/testing, where tests can only depend on system input-output behavior). Audits are evaluations which compare a system's behaviors to expectations about those behaviors, with the goal of establishing that systems behave as intended, and are in this way analogous to traditional software testing methods and serve as a tool for accountability of system behaviors and system-driven outcomes [25]. For this reason, audits are driven by evidence of how a system operates or was constructed, and the audit outcome depends on the reviewability [138] or traceability [7] of the system under examination. Audit regimes drive accountability for system-level behaviors in a variety of domains, and have the potential to be very useful for AI systems, but require the development of better and more complete standards backed by scientific validation that their employment supports substantive system-level goals [139]. But the extent to which having or publicizing audit results leads to system-level changes or accountability for system risks and harms remains unknown [140].

Such evidence can come from system logs and records, but often comes from documentation produced during system development, which might be intended to communicate either facts about system behaviors or intentions about system expectations [141]–[145]. Documentation creates and communicates information about AI systems, but while high-fidelity documentation is a sign of a system operating safely, documentation alone is not a safety intervention: documentation can be out of date, incorrect, or unread, for example. And while checklisting as a tool for baseline task completion or risk information communication is well understood and used in a variety of safety-critical domains [146], there is not yet either an accepted set of baseline tasks or performance measures for AI systems, even in well studied and well circumscribed domains such as aviation. Further, documentation's utility as an intervention has not yet been validated for AI systems, despite the wide array of proposed design interventions based on improved documentation. Some empirical work in this area has begun in the private sector [54], [147], but none has yet focused on AI in safety-critical domains.

Presently, the National Institute of Standards and Technology is developing an AI Risk Management Framework meant to undergird audit and other system management processes for AI systems [5]. While the broad idea of organizational management tools for identifying risks and managing the structures of sociotechnical control around their mitigation is important and connects to management of technology-driven risks in a variety of domains, both safety-critical and not, the present approaches mostly consist of hypotheses about what makes for good risk control. It would be substantially more robust and reliable if these management tools were examined empirically for performance in both identifying and limiting risks, or at least for their performance in identifying failure modes and scenarios (a substantially easier and more concrete task, as losses can be defined and observed while risks are unobservable and require a measure to operationalize). In the safety world, such program effectiveness work is a common and expected part of closing the loop between system outcomes and program operation [148], [149].

The NIST effort is not the only AI audit and risk management effort that attempts to bridge the gap between the need and desire for management and mitigation of AI-driven risk in consequential systems and the lack of standard, accepted processes for said risk management. The most established is probably the Model Risk Management governance framework in the financial industry, governed by the Federal Reserve's supervisory note SR11-7 [150]. This regime ties activities such as impact assessment [151]–[153] to existing organizational governance and risk control methodologies and structures [154]. Other extant audit and risk governance frameworks include the Government Accountability Office's extension of the US Government's "yellow book" auditing standard to cover audits of AI systems [155] and nascent frameworks such as pending revisions to DoD safety standards such as MIL-STD 882E or software airworthiness standards such as the DO-178C assessment standard adopted by the FAA. Medical device assessment standards are also being revised to better apply to AI systems, especially data-driven machine-learning-powered devices. For example, the FDA recently accepted a deep-learning based screening test for diabetic retinopathy as an approved medical device for use outside the direct judgement of a clinician (this framing is a bit deceptive – as a screening test, a positive result indicates a referral to an expert; no diagnosis is made purely autonomously) [156].

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This report analyzes potential hazards associated with AI systems. We taxonomize available mitigations, providing a meta-survey of safety issues and their management. We focus especially on problems related to managing the modeling gap between the world as represented within an AI system and the underlying world in which they system must operate. Additionally, we take a whole-system view of safety, looking to AI tools embodied within systems and the way those systems can fail as a result of AI tool behaviors, rather than simply focusing on improving reliability at the component level.

In general, safety of AI systems is difficult to assess, and many of the AI systems that have achieved high prediction accuracy are not robust with respect to perturbations of system inputs. This is a concern for safety-critical systems, particularly for applications that need to operate in contested environments. Examples include financial applications that operate on the internet and defense applications. In both cases, adversaries are actively and adaptively seeking ways to make the systems fail. The reasons why safety of AI systems is more difficult to assess than for other kinds of systems are discussed in Section III of this report.

Hazards associated with AI systems and known mitigations for those hazards are surveyed in Section IV. There is a great deal of previous work in these areas, and progress has been made. However, at the time of this writing, there are no complete solutions for mitigating the known hazards, and much work remains to be done before such solutions become available.

We conclude that AI systems are often fragile and subject to failures [4] – both our own systems and those of our competitors and adversaries. At the current state of the art, the best opportunities for achieving safe and effective applications of AI are in controlled environments, where closed-world models are applicable and can be verified and validation can be appropriately scoped, in contrast to uncontrolled environments, where open-world models are needed and validation is problematic.

Although much work focuses on improving metrizable properties of AI components, our interest is in failures at the systemic level. For this, it is necessary to connect system level hazards and failures to requirements on component performance. We hypothesize that, while there are several major barriers to AI evaluation, some requiring foundational research to overcome, system safety frameworks are well suited to building these needed connections [157]. The existing work described in this report can then be applied to achieve the identified performance requirements.

This report focuses on short- to medium-term issues and system-level safety concerns. See [158] for a discussion of long-term concerns, most of which are economic and political, with mitigations related to changes in policy and incentives.

B. RECOMMENDATIONS FOR SAFETY ASSESSMENTS

We recommend the following guidelines for design and safety assessment of AI systems.

- Adopt a whole-system approach to safety of systems containing AI technology. At the current time, methods for assessing AI components are immature, emerging approaches are difficult to apply, and are mostly practical and effective only in special cases with particular properties and modest complexity. Rather than relying on robust operation of AI subsystems, we suggest that the larger systems containing those components need fault-tolerant designs that provide fault detection mechanisms and backup systems with alternative solutions to cover cases where the AI components fail. Safety assessments should include safety cases for the effectiveness of the fault detection mechanisms and adequacy of the backup systems. System safety methods are known and likely suited to this task, but future work should explore the extent to which this hypothesis holds.
- In the context of man-machine teaming, where human experts are used to check for possible failures of AI components and mitigate them, we recommend requiring and assessing effectiveness of training on how to respond to possible component failures, assessing interfaces of the systems to check that they support adequate situational awareness for the experts to understand what is going wrong in detected failures and to check that the interfaces provide means for adequate recovery actions. Issues of human comprehension of system actions are rarely issues of explainability or inscrutability, but rather issues of whether humans can effectively integrate AI components into their workflow within whole sytems. Safety assessments of such designs should also evaluate whether the rate of false alarms in the system is sufficiently low that operators will not routinely ignore system alerts.

C. RECOMMENDATIONS FOR FUTURE WORK

We recommend future work on the following aspects of AI safety.

- Test the hypothesis that systems safety methods can be adapted to articulating performance requirements for AI components used in safety-assured systems, rather than focusing on the safety of the components themselves. This may require adapting known safety methodologies such as STPA and FMEA to focus on AI-specific hazard etiologies [56], [157] or inventing specific hazard identification methods [159] that relate to known taxonomies of AI-relevant hazards, such as the one presented in Section IV.
- Find effective methods for validating AI models – checking whether they match the real world. This problem is poorly studied and strongly affects AI system safety assessments. See related discussion in section II.A.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete Problems in AI Safety," *ArXiv160606565 Cs*, Jul. 2016, Accessed: Mar. 07, 2022. [Online]. Available: http://arxiv.org/abs/1606.06565

[2] D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt, "Unsolved Problems in ML Safety," *ArXiv210913916 Cs*, Dec. 2021, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/2109.13916

[3] "Final Report: National Security Commission on Artificial Intelligence." National Security Commission on Artificial Intelligence, Mar. 2021. [Online]. Available: https://www.nscai.gov/

[4] E. Jatho and J. A. Kroll, "Artificial Intelligence: Too Fragile to Fight?," *Proceedings of the United States Naval Institute*, vol. 148/2/1, p. 428.

[5] "AI Risk Management Framework Concept Paper." National Institute of Standards and Technology, Dec. 2021.

[6] "Department of Defense Directive 3000.09: Autonomy in Weapons Systems." Office of the Deputy Secretary of Defense, Nov. 21, 2012.

[7] J. A. Kroll, "Outlining Traceability: A Principle for Operationalizing Accountability in Computing Systems," in *2021 Conference on Fairness, Accountability, and Transparency*, 2021, pp. 1–13.

[8] A. Z. Jacobs, "Measurement as governance in and for responsible AI," *ArXiv210905658 Cs*, Sep. 2021, Accessed: Dec. 22, 2021. [Online]. Available: http://arxiv.org/abs/2109.05658

[9] D. J. Hand, *Measurement: a very short introduction*, First edition. Oxford, United Kingdom: Oxford University Press, 2016.

[10] A. Z. Jacobs and H. Wallach, "Measurement and Fairness," in *ACM Conference on Fairness, Accountability, and Transparency*, ACM, 2021.

[11] H. Levesque, E. Davis, and L. Morgenstern, "The winograd schema challenge," *Thirteen. Int. Conf. Princ. Knowl. Represent. Reason.*.

[12] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, "Language (technology) is power: A critical survey of 'bias' in NLP," *Annu. Meet. Assoc. Comput. Linguist. ACL*, 2020.

[13] S. Gaube *et al.*, "Do as AI say: susceptibility in deployment of clinical decision-aids," *Npj Digit. Med.*, vol. 4, no. 1, p. 31, Dec. 2021, doi: 10.1038/s41746-021-00385-9.

[14] M. Elish and T. Hwang, "Praise the Machine! Punish the Human! The contradictory history of accountability in automated aviation," *Data Soc. Rep.*, 2015.

[15] L. Bainbridge, "Ironies of automation," in *Analysis, design and evaluation of man–machine systems*, Elsevier, 1983, pp. 129–135.

[16] M. C. Elish, “Moral crumple zones: Cautionary tales in human-robot interaction,” *Engag. Sci. Technol. Soc.*, vol. 5, pp. 40–60, 2019.

[17] M. A. Chang, B. Tschaen, T. Benson, and L. Vanbever, “Chaos Monkey: Increasing SDN Reliability through Systematic Network Destruction,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, London United Kingdom, Aug. 2015, pp. 371–372. doi: 10.1145/2785956.2790038.

[18] D. Vaughan, *The Challenger launch decision: Risky technology, culture, and deviance at NASA*. University of Chicago press, 1996.

[19] M. M. Malik, “A Hierarchy of Limitations in Machine Learning,” *ArXiv200205193 Cs Econ Math Stat*, Feb. 2020, Accessed: Oct. 08, 2021. [Online]. Available: http://arxiv.org/abs/2002.05193

[20] S. Nemati, A. Holder, F. Razmi, M. D. Stanley, G. D. Clifford, and T. G. Buchman, “An Interpretable Machine Learning Model for Accurate Prediction of Sepsis in the ICU:,” *Crit. Care Med.*, vol. 46, no. 4, pp. 547–553, Apr. 2018, doi: 10.1097/CCM.0000000000002936.

[21] C. Barton *et al.*, “Evaluation of a machine learning algorithm for up to 48-hour advance prediction of sepsis using six vital signs,” *Comput. Biol. Med.*, vol. 109, pp. 79–84, Jun. 2019, doi: 10.1016/j.compbiomed.2019.04.027.

[22] L. M. Fleuren *et al.*, “Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy,” *Intensive Care Med.*, vol. 46, no. 3, pp. 383–400, Mar. 2020, doi: 10.1007/s00134-019-05872-y.

[23] A. Wong *et al.*, “External Validation of a Widely Implemented Proprietary Sepsis Prediction Model in Hospitalized Patients,” *JAMA Intern. Med.*, vol. 181, no. 8, p. 1065, Aug. 2021, doi: 10.1001/jamainternmed.2021.2626.

[24] J. Yang, L. Karstens, C. Ross, and A. Yala, “AI Gone Astray: Technical Supplement.” arXiv, Feb. 28, 2022. Accessed: Dec. 08, 2022. [Online]. Available: http://arxiv.org/abs/2203.16452

[25] J. A. Kroll *et al.*, “Accountable Algorithms,” *Univ. Pa. Law Rev.*, vol. 165, no. 3, pp. 633–705, 2017.

[26] J. Buolamwini and T. Gebru, “Gender shades: Intersectional accuracy disparities in commercial gender classification,” in *Conference on Fairness, Accountability and Transparency*, 2018, pp. 77–91.

[27] P. J. Bickel, E. A. Hammel, and J. W. O’Connell, “Sex Bias in Graduate Admissions: Data from Berkeley: Measuring bias is harder than is usually assumed, and the evidence is sometimes contrary to expectation.,” *Science*, vol. 187, no. 4175, pp. 398–404, Feb. 1975, doi: 10.1126/science.187.4175.398.

[28] J. Burn-Murdoch, “Germany’s Election and the Trouble with Correlation,” Oct. 02, 2017.

[29] J. M. Kleinberg, “An impossibility theorem for clustering,” in *Advances in neural information processing systems*, 2003, pp. 463–470.

[30] J. Pearl, *Causality*. Cambridge university press, 2009.

[31] S. A. Snook, *Friendly Fire: The Accidental Shootdown of U.S. Black Hawks over Northern Iraq*. Princeton: Princeton University Press, 2000. doi: 10.1515/9781400840977.

[32] M. L. Gray and S. Suri, *Ghost work: how to stop Silicon Valley from building a new global underclass*. Boston: Houghton Mifflin Harcourt, 2019.

[33] D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian, "Runaway Feedback Loops in Predictive Policing," in *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, 2018, vol. 81, pp. 160–171. [Online]. Available: http://proceedings.mlr.press/v81/ensign18a.html

[34] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *ArXiv14126572 Cs Stat*, Mar. 2015, Accessed: Oct. 08, 2021. [Online]. Available: http://arxiv.org/abs/1412.6572

[35] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbrucken, Mar. 2016, pp. 372–387. doi: 10.1109/EuroSP.2016.36.

[36] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial Examples Are Not Bugs, They Are Features," *Proc. Neural Inf. Process. Symp.*, 2019, Accessed: Dec. 30, 2021. [Online]. Available: http://arxiv.org/abs/1905.02175

[37] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1633–45.

[38] A. Barton, E. Jatho, and V. Berzins, "Defending Against Adversarial Examples in Deep Neural Network Classifiers," Naval Postgraduate School, NPS-CS-21-002, Dec. 2021.

[39] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet Classifiers Generalize to ImageNet?," *Proc. Int. Conf. Mach. Learn.*, 2019, Accessed: Dec. 30, 2021. [Online]. Available: http://arxiv.org/abs/1902.10811

[40] A. Shamir, O. Melamed, and O. BenShmuel, "The Dimpled Manifold Model of Adversarial Examples in Machine Learning," *ArXiv Prepr. ArXiv210610151*.

[41] I. D. Raji, E. M. Bender, A. Paullada, E. Denton, and A. Hanna, "AI and the Everything in the Whole Wide World Benchmark," *ArXiv211115366 Cs*, Nov. 2021, Accessed: Dec. 30, 2021. [Online]. Available: http://arxiv.org/abs/2111.15366

[42] R. Schwartz, L. Down, A. Jonas, and E. Tabassi, "Special Publication 1270: A Proposal for Identifying and Managing Bias in Artificial Intelligence Systems." National Institute of Standards and Technology.

[43] B. Friedman and H. Nissenbaum, "Bias in computer systems," *ACM Trans. Inf. Syst. TOIS*, vol. 14, no. 3, pp. 330–347, 1996.

[44] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson Education Limited, 2016.

[45] K. Lum and W. Isaac, “To predict and serve?,” *Significance*, vol. 13, no. 5, pp. 14–19, 2016.

[46] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, “Machine Bias,” *ProPublica*, May 2016.

[47] S. Corbett-Davies, E. Pierson, A. Feller, and S. Goel, *A computer program used for bail and sentencing decisions was labeled biased against blacks. It’s actually not that clear.* 2016.

[48] I. Ajunwa, “The Auditing Imperative for Automated Hiring,” *Harv. J. Law Technol.*, vol. 34, 2021.

[49] M. Raghavan, S. Barocas, J. Kleinberg, and K. Levy, “Mitigating bias in algorithmic hiring: Evaluating claims and practices,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 469–481.

[50] D. K. Mulligan, J. A. Kroll, N. Kohli, and R. Y. Wong, “This Thing Called Fairness: Disciplinary Confusion Realizing a Value in Technology,” *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, p. 119, 2019.

[51] M. K. Lee *et al.*, “WeBuildAI: Participatory framework for algorithmic governance,” *Proc. ACM Hum.-Comput. Interact.*, vol. 3, no. CSCW, pp. 1–35, 2019.

[52] F. Bonchi, C. Castillo, and S. Hajian, *Algorithmic bias: from discrimination discovery to fairness-aware data mining*. 2016.

[53] M. Hardt, S. Barocas, and A. Narayanan, *Fairness in Machine Learning: Limitations and Opportunities*. Online, 2018. [Online]. Available: https://mrtz.org/preview

[54] K. Holstein, J. W. Vaughan, H. Daumé III, M. Dudík, and H. Wallach, “Improving fairness in machine learning systems: What do industry practitioners need?,” *Proc. ACM Conf. Comput.-Hum. Interact. CHI*, 2019.

[55] M. Madaio, L. Egede, H. Subramonyam, J. W. Vaughan, and H. Wallach, “Assessing the Fairness of AI Systems: AI Practitioners’ Processes, Challenges, and Needs for Support,” *Proc. ACM Conf. Comput. Support. Coop. Work CSCW*, 2022, Accessed: Dec. 31, 2021. [Online]. Available: http://arxiv.org/abs/2112.05675

[56] S. Rismani *et al.*, “From plane crashes to algorithmic harm: applicability of safety engineering frameworks for responsible ML.” arXiv, Oct. 05, 2022. Accessed: Dec. 04, 2022. [Online]. Available: http://arxiv.org/abs/2210.03535

[57] I. Kohler-Hausmann, “Eddie Murphy and the dangers of counterfactual causal thinking about detecting racial discrimination,” *Nw UL Rev*, vol. 113, p. 1163, 2018.

[58] A. Caliskan, J. J. Bryson, and A. Narayanan, “Semantics derived automatically from language corpora contain human-like biases,” *Science*, vol. 356, no. 6334, pp. 183–186, Apr. 2017, doi: 10.1126/science.aal4230.

[59] "Eye In the Sky: DOD Announces AI Challenge." Defense Innovation Unit. [Online]. Available: https://www.defense.gov/News/News-Stories/Article/Article/1934806/eye-in-the-sky-dod-announces-ai-challenge/

[60] S. Corbett-Davies and S. Goel, "The measure and mismeasure of fairness: A critical review of fair machine learning," *ArXiv Prepr. ArXiv180800023*, 2018.

[61] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under Concept Drift: A Review," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2018, doi: 10.1109/TKDE.2018.2876857.

[62] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Apr. 2014, doi: 10.1145/2523813.

[63] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift," *J. Mach. Learn. Res.*, vol. 22, no. 98, pp. 1–76, 2021.

[64] S. Shalev-Shwartz, "Online Learning and Online Convex Optimization," *Found. Trends® Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011, doi: 10.1561/2200000018.

[65] R. French, "Catastrophic forgetting in connectionist networks," *Trends Cogn. Sci.*, vol. 3, no. 4, pp. 128–135, Apr. 1999, doi: 10.1016/S1364-6613(99)01294-2.

[66] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proc. Natl. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, doi: 10.1073/pnas.1611835114.

[67] M. Cummings, "Automation bias in intelligent time critical decision support systems," in *AIAA 1st Intelligent Systems Technical Conference*, 2004, p. 6313.

[68] N. G. Leveson, *Engineering a safer world: Systems thinking applied to safety*. The MIT Press, 2016.

[69] M. L. Cummings and S. Guerlain, "Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles," *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 49, no. 1, pp. 1–15, Feb. 2007, doi: 10.1518/001872007779598109.

[70] Goldenfein, Jake; Mulligan, Deirdre K.; Nissenbaum, Helen; Ju, Wendy, "Through the Handoff Lens: Competing Visions of Autonomous Futures," 2021, doi: 10.15779/Z38CR5ND0J.

[71] D. A. Norman, "The 'problem ' with automation: inappropriate feedback and interaction, not 'over-automation,'" *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, vol. 327, no. 1241, pp. 585–593, Apr. 1990, doi: 10.1098/rstb.1990.0101.

[72] S. W. A. Dekker and D. D. Woods, "MABA-MABA or Abracadabra? Progress on Human-Automation Co-ordination," *Cogn. Technol. Work*, vol. 4, no. 4, pp. 240–244, Nov. 2002, doi: 10.1007/s101110200022.

[73] B. Palmer, *Understanding Air France 447*. 2013.

[74] "Collision of Two Washington Metropolitan Area Transit Authority Metrorail Trains Near Fort Totten Station." National Transportation Safety Board.

[75] P. J. Denning and J. Arquilla, “The context problem in artificial intelligence,” *Commun. ACM*, vol. 65, no. 12, pp. 18–21, Dec. 2022, doi: 10.1145/3567605.

[76] G. McGraw, H. Figueroa, V. Shepardson, and R. Bonett, “An Architectural Risk Analysis of Machine Learning Systems: Towards More Secure Machine Learning.” Berryville Institute of Machine Learning.

[77] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, “SoK: Security and Privacy in Machine Learning,” in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, London, Apr. 2018, pp. 399–414. doi: 10.1109/EuroSP.2018.00035.

[78] J. Steinhardt, P. W. Koh, and P. Liang, “Certified Defenses for Data Poisoning Attacks,” *Proc. Neural Inf. Process. Symp.*, 2017.

[79] M. Fredrikson, S. Jha, and T. Ristenpart, “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver Colorado USA, Oct. 2015, pp. 1322–1333. doi: 10.1145/2810103.2813677.

[80] P. Grubbs, M.-S. Lacharite, B. Minaud, and K. G. Paterson, “Learning to Reconstruct: Statistical Learning Theory and Encrypted Database Attacks,” in *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 1067–1083. doi: 10.1109/SP.2019.00030.

[81] K. Thompson, “Reflections on trusting trust,” *Commun. ACM*, vol. 27, no. 8, pp. 761–763, Aug. 1984, doi: 10.1145/358198.358210.

[82] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum, “Privacy and contextual integrity: Framework and applications,” in *Security and Privacy, 2006 IEEE Symposium on*, 2006, pp. 15-pp.

[83] T. Carter, J. A. Kroll, and J. Bret Michael, “Lessons Learned From Applying the NIST Privacy Framework,” *IT Prof.*, vol. 23, no. 4, pp. 9–13, Jul. 2021, doi: 10.1109/MITP.2021.3086916.

[84] R. M. O’Keefe and D. E. O’Leary, “Expert system verification and validation: a survey and tutorial,” *Artif. Intell. Rev.*, vol. 7, no. 1, pp. 3–42, Feb. 1993, doi: 10.1007/BF00849196.

[85] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artif. Intell.*, vol. 267, pp. 1–38, 2019.

[86] M. Landry, J.-L. Malouin, and M. Oral, “Model validation in operations research,” *Eur. J. Oper. Res.*, vol. 14, no. 3, pp. 207–220, Nov. 1983, doi: 10.1016/0377-2217(83)90257-6.

[87] J. Rushby, “Quality Measures and Assurance for AI Software,” *NASA Contract. Rep. 4187*.

[88] C. Perrow, *Normal accidents: Living with high risk technologies*. New York: Basic Books, 1984.

[89] N. Fulton and A. Platzer, “Safe Reinforcement Learning via Formal Methods: Toward Safe Control Through Proof and Learning,” *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, Accessed: Oct. 08, 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/12107

[90] H.-D. Tran, F. Cai, M. L. Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, “Safety Verification of Cyber-Physical Systems with Reinforcement Learning Control,” *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, pp. 1–22, Oct. 2019, doi: 10.1145/3358230.

[91] G. C. Bowker and S. L. Star, *Sorting things out: Classification and its consequences*. MIT press, 2000.

[92] C. Szegedy *et al.*, “Intriguing properties of neural networks,” *ArXiv13126199 Cs*, Feb. 2014, Accessed: Feb. 04, 2022. [Online]. Available: http://arxiv.org/abs/1312.6199

[93] S. Gowal *et al.*, “On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models,” *ArXiv181012715 Cs Stat*, Aug. 2019, Accessed: Feb. 04, 2022. [Online]. Available: http://arxiv.org/abs/1810.12715

[94] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards Deep Learning Models Resistant to Adversarial Attacks,” *Int. Conf. Learn. Represent.*, 2018, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/1706.06083

[95] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, and M. J. Kochenderfer, “Algorithms for Verifying Deep Neural Networks,” *Found. Trends® Optim.*, vol. 4, no. 3–4, pp. 244–404, 2021, doi: 10.1561/2400000035.

[96] C.-H. Cheng, G. Nührenberg, and H. Ruess, “Maximum Resilience of Artificial Neural Networks,” in *Automated Technology for Verification and Analysis*, vol. 10482, D. D’Souza and K. Narayan Kumar, Eds. Cham: Springer International Publishing, 2017, pp. 251–268. doi: 10.1007/978-3-319-68167-2_18.

[97] V. Tjeng, K. Xiao, and R. Tedrake, “Evaluating Robustness of Neural Networks with Mixed Integer Programming,” *ArXiv171107356 Cs*, Feb. 2019, Accessed: Feb. 25, 2022. [Online]. Available: http://arxiv.org/abs/1711.07356

[98] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, “Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks,” in *Computer Aided Verification*, vol. 10426, R. Majumdar and V. Kunčak, Eds. Cham: Springer International Publishing, 2017, pp. 97–117. doi: 10.1007/978-3-319-63387-9_5.

[99] R. Ehlers, “Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks,” in *Automated Technology for Verification and Analysis*, vol. 10482, D. D’Souza and K. Narayan Kumar, Eds. Cham: Springer International Publishing, 2017, pp. 269–286. doi: 10.1007/978-3-319-68167-2_19.

[100] L. Pulina and A. Tacchella, “An Abstraction-Refinement Approach to Verification of Artificial Neural Networks,” in *Computer Aided Verification*, vol. 6174, T. Touili, B. Cook, and P. Jackson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 243–257. doi: 10.1007/978-3-642-14295-6_24.

[101] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, “Efficient Machine Learning for Big Data: A Review,” *Big Data Res.*, vol. 2, no. 3, pp. 87–93, Sep. 2015, doi: 10.1016/j.bdr.2015.04.001.

[102] P. Henriksen and A. Lomuscio, “Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search,” *Eur. Conf. AI*, vol. 325, no. ECAI 2020, 2020.

[103] R. Bunel, I. Turkaslan, P. H. S. Torr, P. Kohli, and M. P. Kumar, “A Unified View of Piecewise Linear Neural Network Verification,” *Proc. Neural Inf. Process. Symp.*, vol. 2018, 2018.

[104] C. Urban and A. Miné, “A Review of Formal Methods applied to Machine Learning,” *ArXiv210402466 Cs*, Apr. 2021, Accessed: Mar. 03, 2022. [Online]. Available: http://arxiv.org/abs/2104.02466

[105] F. A. Batarseh, L. Freeman, and C.-H. Huang, “A survey on artificial intelligence assurance,” *J. Big Data*, vol. 8, no. 1, p. 60, Dec. 2021, doi: 10.1186/s40537-021-00445-7.

[106] R. Ashmore, R. Calinescu, and C. Paterson, “Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges,” *ACM Comput. Surv.*, vol. 54, no. 5, pp. 1–39, Jun. 2021, doi: 10.1145/3453444.

[107] N. Rajabli, F. Flammini, R. Nardone, and V. Vittorini, “Software Verification and Validation of Safe Autonomous Cars: A Systematic Literature Review,” *IEEE Access*, vol. 9, pp. 4797–4819, 2021, doi: 10.1109/ACCESS.2020.3048047.

[108] F. Pasquale, *The black box society: The secret algorithms that control money and information*. Harvard University Press, 2015.

[109] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, “XAI—Explainable artificial intelligence,” *Sci. Robot.*, vol. 4, no. 37, p. eaay7120, Dec. 2019, doi: 10.1126/scirobotics.aay7120.

[110] D. A. Broniatowski, “Psychological Foundations of Explainability and Interpretability in Artificial Intelligence,” National Institute of Standards and Technology, Apr. 2021. doi: 10.6028/NIST.IR.8367.

[111] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney NSW Australia, Aug. 2015, pp. 1721–1730. doi: 10.1145/2783258.2788613.

[112] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, “Trends and Trajectories for Explainable, Accountable and Intelligible Systems,” *Proc. Int. Conf. Hum. Factors Comput. Syst. CHI*, 2018.

[113] A. D. Selbst and S. Barocas, “The intuitive appeal of explainable machines,” *Fordham Rev*, vol. 87, 2018.

[114] E. Rader, K. Cotter, and J. Cho, “Explanations as Mechanisms for Supporting Algorithmic Transparency,” *Proc. Int. Conf. Hum. Factors Comput. Syst. CHI*, 2018.

[115] U. Bhatt *et al.*, “Explainable machine learning in deployment,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, Barcelona Spain, Jan. 2020, pp. 648–657. doi: 10.1145/3351095.3375624.

[116] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black-box Models Explainable*. Online: https://christophm.github.io/interpretable-ml-book/, 2018.

[117] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” *ArXiv Prepr. ArXiv170208608*, 2017.

[118] C. Rudin, “Algorithms for interpretable machine learning,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1519–1519.

[119] E. Tjoa and C. Guan, “A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Nov. 2021, doi: 10.1109/TNNLS.2020.3027314.

[120] D. Gunning, “Machine Common Sense Concept Paper,” *ArXiv181007528 Cs*, Oct. 2018, Accessed: Mar. 22, 2022. [Online]. Available: http://arxiv.org/abs/1810.07528

[121] R. I. Cook, “How complex systems fail,” *Cogn. Technol. Lab. Univ. Chic. Chic. IL*, 1998.

[122] B. Biggio *et al.*, “Evasion Attacks against Machine Learning at Test Time,” in *Advanced Information Systems Engineering*, vol. 7908, C. Salinesi, M. C. Norrie, and Ó. Pastor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 387–402. doi: 10.1007/978-3-642-40994-3_25.

[123] K. Levchenko *et al.*, “Click Trajectories: End-to-End Analysis of the Spam Value Chain,” in *2011 IEEE Symposium on Security and Privacy*, Berkeley, CA, May 2011, pp. 431–446. doi: 10.1109/SP.2011.24.

[124] E. Hüllermeier and W. Waegeman, “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods,” *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, Mar. 2021, doi: 10.1007/s10994-021-05946-3.

[125] S. Sankararaman and S. Mahadevan, “Model validation under epistemic uncertainty,” *Reliab. Eng. Syst. Saf.*, vol. 96, no. 9, pp. 1232–1241, Sep. 2011, doi: 10.1016/j.ress.2010.07.014.

[126] M. Bunge, *Causality and Modern Science*, 1st ed. Routledge, 2017. doi: 10.4324/9781315081656.

[127] B. Schölkopf, “Causality for Machine Learning,” in *Probabilistic and Causal Inference*, 1st ed., H. Geffner, R. Dechter, and J. Y. Halpern, Eds. New York, NY, USA: ACM, 2022, pp. 765–804. doi: 10.1145/3501714.3501755.

[128] S. Mullainathan and J. Spiess, “Machine Learning: An Applied Econometric Approach,” *J. Econ. Perspect.*, vol. 31, no. 2, pp. 87–106, May 2017, doi: 10.1257/jep.31.2.87.

[129] L. Bottou *et al.*, “Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising,” *J. Mach. Learn. Res.*, vol. 14, no. 65, pp. 3207–3260, 2013.

[130] A. Kasirzadeh and A. Smart, “The Use and Misuse of Counterfactuals in Ethical Machine Learning,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 228–236.

[131] G. Falco *et al.*, “Governing AI safety through independent audits,” *Nat. Mach. Intell.*, vol. 3, no. 7, pp. 566–571, Jul. 2021, doi: 10.1038/s42256-021-00370-7.

[132] C. Wilson *et al.*, “Building and Auditing Fair Algorithms: A Case Study in Candidate Screening,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event Canada, Mar. 2021, pp. 666–677. doi: 10.1145/3442188.3445928.

[133] M. A. Bashir, S. Arshad, and C. Wilson, “Recommended For You: A First Look at Content Recommendation Networks,” in *Proceedings of the 2016 Internet Measurement Conference*, 2016.

[134] L. Chen and C. Wilson, “Observing algorithmic marketplaces in-the-wild,” *ACM SIGecom Exch.*, vol. 15, no. 2, pp. 34–39, 2017.

[135] A. Hannak *et al.*, “Measuring personalization of web search,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[136] L. Chen, R. Ma, A. Hannák, and C. Wilson, “Investigating the Impact of Gender on Rank in Resume Search Engines,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018.

[137] A. Hannak, G. Soeller, D. Lazer, A. Mislove, and C. Wilson, “Measuring price discrimination and steering on e-commerce web sites,” in *Proceedings of the 2014 conference on internet measurement conference*, pp. 305–318.

[138] J. Cobbe, M. S. A. Lee, and J. Singh, “Reviewable Automated Decision-Making: A Framework for Accountable Algorithmic Systems,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event Canada, Mar. 2021, pp. 598–609. doi: 10.1145/3442188.3445921.

[139] I. D. Raji *et al.*, “Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing,” *ACM Conf. Fairness Account. Transpar.*, 2020.

[140] I. D. Raji and J. Buolamwini, “Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products,” in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, Honolulu HI USA, Jan. 2019, pp. 429–435. doi: 10.1145/3306618.3314244.

[141] T. Gebru *et al.*, “Datasheets for datasets,” *ArXiv Prepr. ArXiv180309010*, 2018.

[142] M. Mitchell *et al.*, “Model cards for model reporting,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 220–229.

[143] S. Holland, A. Hosny, S. Newman, J. Joseph, and K. Chmielinski, “The dataset nutrition label: A framework to drive higher data quality standards,” *ArXiv Prepr. ArXiv180503677*, 2018.

[144] Partnership on AI, *Annotation and Benchmarking on Understanding and Transparency of Machine Learning Lifecycles (ABOUT ML)*. 2019. [Online]. Available: https://partnershiponai.com/about-ml

[145] E. M. Bender and B. Friedman, “Data statements for natural language processing: Toward mitigating system bias and enabling better science,” *Trans. Assoc. Comput. Linguist.*, vol. 6, pp. 587–604, 2018.

[146] A. Gawande, *The Checklist Manifesto: How to Get Things Right*. Metropolitan Books, 2009.

[147] M. A. Madaio, L. Stark, J. Wortman Vaughan, and H. Wallach, “Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Honolulu HI USA, Apr. 2020, pp. 1–14. doi: 10.1145/3313831.3376445.

[148] S. Piric, R. J. De Boer, A. Roelen, N. Karanikas, and S. Kaspers, “How does aviation industry measure safety performance Current practice and limitations,” *Int. J. Aviat. Manag.*, vol. 4, no. 3, p. 224, 2019, doi: 10.1504/IJAM.2019.10019874.

[149] P. O’Connor, A. O’Dea, Q. Kennedy, and S. E. Buttrey, “Measuring safety climate in aviation: A review and recommendations for the future,” *Saf. Sci.*, vol. 49, no. 2, pp. 128–138, Feb. 2011, doi: 10.1016/j.ssci.2010.10.001.

[150] Division of Banking Supervision and Regulation, *SR 11-7: Guidance on Model Risk Management*. Board of Governors of the Federal Reserve System, 2011.

[151] D. Reisman, J. Schultz, K. Crawford, and M. Whittaker, *Algorithmic Impact Assessements: A Practical Framework for Public Agency Accountability*. AI Now Institute, 2018.

[152] J. Metcalf, E. Moss, E. A. Watkins, R. Singh, and M. C. Elish, “Algorithmic Impact Assessments and Accountability: The Co-construction of Impacts,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event Canada, Mar. 2021, pp. 735–746. doi: 10.1145/3442188.3445935.

[153] E. Moss, E. A. Watkins, R. Singh, M. C. Elish, and J. Metcalf, “Assembling Accountability: Algorithmic Impact Assessment for the Public Interest,” Data & Society Research Institute, Jun. 2021.

[154] H. Davies and M. Zhivitskaya, “Three Lines of Defence: A Robust Organising Framework, or Just Lines in the Sand?,” *Glob. Policy*, vol. 9, pp. 34–42, Jun. 2018, doi: 10.1111/1758-5899.12568.

[155] T. Ariga, T. Persons, and S. Sanford, “Artificial Intelligence: An Accountability Framework for Federal Agencies and Other Entities,” Government Accountability Office, GAO-21-519SP, Jun. 2021.

[156] M. Ratner, “FDA backs clinician-free AI imaging diagnostic tools,” *Nat. Biotechnol.*, vol. 36, no. 8, pp. 673–674, Sep. 2018, doi: 10.1038/nbt0818-673a.

[157] E. W. Jatho III, L. O. Mailloux, S. Rismani, E. D. Williams, and J. A. Kroll, “System Safety Engineering for Social and Ethical ML Risks: A Case Study.” arXiv, Nov. 08, 2022. Accessed: Dec. 07, 2022. [Online]. Available: http://arxiv.org/abs/2211.04602

[158] E. Brynjolfsson, “The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence,” *Daedalus*, vol. 151, no. 2, pp. 272–287, May 2022, doi: 10.1162/daed_a_01915.

[159] H. Khlaaf, P. Mishkin, J. Achiam, G. Krueger, and M. Brundage, “A Hazard Analysis Framework for Code Synthesis Large Language Models.” arXiv, Jul. 25, 2022. Accessed: Dec. 07, 2022. [Online]. Available: http://arxiv.org/abs/2207.14157

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Research Sponsored Programs Office, Code 41
   Naval Postgraduate School
   Monterey, CA 93943

4. Commander, Naval Air Systems Command (NAVAIR)
   Patuxent River, Maryland