SYM-AM-21-086

EXCERPT FROM THE PROCEEDINGS OF THE EIGHTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

Increasing Confidence in Machine Learned (ML) Functional Behavior during Artificial Intelligence (AI) Development using Training Data Set Measurements

May 11–13, 2021

Published: May 10, 2021

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School

Increasing Confidence in Machine Learned (ML) Functional Behavior during Artificial Intelligence (AI) Development using Training Data Set Measurements

Bruce Nagy—is a Research Engineer at the Naval Air Warfare Center, Weapons Division at China Lake. His research focuses on advanced game theory techniques, artificial intelligence and machine learning applications for tactical decision aids. Mr. Nagy has earned four degrees: one in mathematics, two in electrical engineering, and one in biology from The Citadel and Naval Postgraduate School. He led the development of advanced algorithms and metrics that resolved national defense issues in satellite communications for DoD. At UCLA during postgraduate work, he investigated modeling brain stem communication with muscle groups at the cellular level, in cooperation with NIH.

Abstract

Both the commercial world and Department of Defense (DoD) are challenged with system safety issues when dealing with Machine Learned (ML)/Artificial Intelligence (AI) deployed products. DoD has a more severe issue when deploying weapons that could unintentionally harm groups of people and property. Commercial manufacturers are motivated by profit, while DoD is motivated by defense readiness. Both are in a race and can suffer the consequences from focusing too much on the finish line. Establishing formal oversight ensures safe algorithm performance. This paper presents a measurement approach that scrutinizes the quality and quantity of training data used when developing ML/AI algorithms. Measuring quality and quantity of training data increases confidence in how the algorithm will perform in a "realistic" operational environment. Combining modality with measurements determines: (1) how to curate data to support a realistic deployed environment; (2) what attributes take priority during training to ensure robust composition of the data; and (3) how attribute prioritization is reflected in size of the training set. The measurements provide a greater understanding of the operational environment, taking into account issues that result when missing and/or sparse data occur, as well as how data sources supply input to the algorithm during deployment.

Introduction

As opposed to traditional software development techniques, Machine Learned (ML)/Artificial Intelligence (AI) created functions have models that are configured using training data sets. Traditional code is used to manage the training process. Training sets are comprised of a combination of attributes, sometimes called features. When we refer to a feature within an image, we are describing a piece of information contained in the content of the image. In this case, the feature describes a certain region of the image, which has certain properties as opposed to another popular definition of a feature as a single pixel in an image. The aggregation of attributes can be contained in one source, e.g., a camera taking a facial picture, or from many sources, e.g., various sensor inputs, such as radar and communication links. In this paper, we will distinguish whether attributes are generated from one or multiple sources based on their modality. As will be described, understanding the type of modality and creating training data sets with the proper quality and quantity of instances/samples to replicate the variation, anomalies and noise experienced during deployment is key to algorithm behavioral confidence.

No Warning Labels

The first woman killed by an autonomous driven car (Schmelzer 2019) provided a reality check for the reliability of AI performance in a deployed environment. Interestingly, 1896 is when the first person was killed by a human driver. Who' was at fault? It was determined to be the driver. When an autonomous system makes a mistake, is it the car or the driver (Gurney 2013) that is at fault? There are many factory recalls of faulty mechanisms in cars, like brakes. Is it any

different with AI software systems? The objective for many car developers is how safe can they make a car using autonomy as compared to other manufacturers (Griffith E 2016). That is their key to their advertising to create acceptance and sales from the consumer.

Elon Musk states a major concern in that AI systems might be developed in secret (Etherington 2012), thereby limiting oversight. For example, Microsoft has exclusive rights to OpenAI's text generation software (Hamilton 2020). This goes against the initial policy by Elon Musk as one of the founders of OpenAI with the goal of developing open source technology. Over the last decade and beyond, the primary motivator for car companies has been money. Over a 20-month period, a company producing technology for driverless cars was involved with 18 accidents (Wiggers 2020**)**. This company declined to support a conglomerate of major automotive developers focused on "safety first" guiding principles (Wiggers 2019**)** in autonomous vehicles. Instead, the company publicly stated that they support laws and regulations. From a legal standpoint, it is quite uncertain that existing laws will apply (Moses LB 2007). Because of that, car manufacturers may not have the proper incentives to develop safe systems (Cooter 2000). Even with this company's public being against a proactive safety focus, even with 18-accidents in 20 months, they were still able to raise over 3 billion dollars. Some legal thoughts support limited regulation, but with the caveat of incentivizing commercial manufacturers to only develop beneficial/useful AI (McGinnis JO 2010). For better or worse, discussions about economics, law, and philosophy (Russell 2015) are attempting to shape the answer to what is beneficial/useful. How to intrinsically motivate (Baum 2017) developers to create beneficial AI? The challenge is that people justify actions based on needs (Kunda 1990). Commercial manufactures must support their bottom-line, whereas DoD has a different set of goals.

The DoD Unique AI Challenge – It's Secret!

DoD has a different set of standards with regard to what is beneficial as compared to commercial needs. Yet, there are many things we can learn from industry. Certainly, DoD cannot afford an international incident regarding an autonomous system, especially a series of incidents that occur over a 20 month period. The significant challenge is that DoD must develop solutions in a sheltered and isolated environment, even from other classified projects. Even open source based oversight is limited. That is the reason system safety organizations are so vital in DoD and must have standards, measurements, policies and procedures to support their effort. Whether in the commercial world or DoD, AI functionality is considered to be unpredictable, unexplainable and goal uncertain (Yampolskiy, 2020). When we talk about AI safety issues for naval weapon systems, this has not typically included adversarial attacks that might affect functional performance. Given this perspective, AI adversarial network attacks using techniques like DeepFakes, putting an image/video into another image/video for mis-categorization (Chauhan 2018), was not included this research, but may be considered for future investigation. Unpredictable, unexplainable and goal uncertain is still an significant issue with AI deployed technology, even when developers are motivated and doing their best (Deci 1971, Krantz 2008). Even the best is still resulting in 18 incidents in 20 months.

A major challenge that both DoD and commercial manufactures face is a race to the finish line (Armstrong 2016) approach to development. Is there something to be learned from the nuclear arms race? The obvious lesson is that we need oversight in the early stages of development (Borrie 2014). AI may have the same dramatic effect, as did the nuclear arms race. Consider the issues of putting military drones and weaponry under the full control of AI systems (Bohannon 2015). Now consider Murphy's Law, "Anything that can go wrong, will." When it comes to what we expect computers to do and what they will actually do, especially when development gets more complex, unwanted incidents are more likely to occur (Joy

2000). Note that the majority of research that was been initiated over a decade ago involving robotic “decisions” and actions was being funded by the military (Lin 2011).

What DoD must ask is, “Can we deploy AI in safety critical functions, i.e., AI-enabled weapons acting autonomously?” The challenge to answering this question is in determining if an AI system can ever be “fixed”, become more reliable, to support safety needs, like brakes in a car.

For both commercial and government AI development, the need for safety standards is becoming more prominent (Ozlati 2017). The federal government has taken the initiative. National Institute of Standards and Technology (NIST) is focused on creating standards that provide oversight for AI development. In their 52 page report (NIST 2019), one of the nine areas of focus is on metrics. This paper is offered for consideration to be included in the NIST standards for AI development with regard to measuring the quantity/size and quality/composition of training data.

Navy System Safety for Weapons Deployment

In order to overcome the unique challenges of ensuring there is adequate safety and security in naval ordnance, the Naval Ordnance Safety and Security Activity (NOSSA) formed. NOSSA, the funding organization for this research, recognized the AI system safety might require a special set of policies, guidelines and metrics. Their concern was that ML/AI algorithms could not be analyzed using traditional hazard analyses approaches (MIL-STD 882E), nor would Federal Aviation Administration rigor guidelines (DO-178C) be adequate. NOSSA wanted to investigate requirements for unique analysis specific to AI development in military systems (Joint SSSEH v1.0). NOSSA also wanted to investigate if any new methodologies were needed to conduct adequate hazard analysis for AI deployed weapon systems (JS-SSA-IF Rev. A).

This research was motivated based on the six critical reasons why the Navy needs to establish measurable confidence in Machine Learned algorithms being deployed in weapons systems:

1. We cannot and should not expect the warfighter to accept and use AI as a social norm (Lapinski 2005), even when the best explainable AI techniques are available, without first having our Acquisition Community measurably establish confidence in Machine Learned algorithms being deployed in realistic operational environments.

2. Acquisition communities cannot identify and certify operational constraints of an ML algorithm for deployment without having confidence in the training data quality, including any negative side effects (Everitt 2018), that might result from the training process.

3. DoD Acquisition communities are limited when following commercial system safety guidelines because the commercial world does not have the same rigor requirements for ensuring AI functional behavior. Commercial manufacturers are driven by profit and may suffer from objective reasoning (Lewandowsky 2015) associated with the conflicting motivation to emphasize safety issues might result in lowering sales.

4. AI upgrades to Navy programs of record that were initially developed following a Capability Maturity Model for traditional software development (Shneiderman 2020) currently exclude ML/AI development differences. Acquisition communities need support and oversight to fill this gap.

5. It is imperative that “Speed to the fleet” deployment of AI systems must overcome their motivational limitations and consider safety impacts of AI using planning, oversight and continuous monitoring by knowledgeable review boards that includes retrospective analysis of disasters (Shneiderman 2016).

6. Navy Weapon System Explosive Safety Review Board (WSESRB) and other approval oversight authorities are limited in their assessment without adequate guidance and tools (Porter 2020, Jones 2019). Guidance and tools need to be a priority in DoD budgets.

AI has a potential of creating a technology leap (Eden 2013). That potential leap, especially when dealing with weapon systems, needs scrutiny. This scrutiny focuses on the specificity of the composition and size of the training data. This research will describe the needed scrutiny by oversight groups can use to increase safety and confidence in the deployment of AI functions.

The AI Acquisition Paradigm

An ML/AI function is selected because it can handle “noisy” inputs and still make a decision as to category or value of the output, the former being categorization and the latter being regression. The success rate of an ML/AI function is the primary measurement, but success is limited to the quality and quantity data input used to train the algorithm. Because of this dependency, “garbage in, garbage out” becomes a determining factor in the capability of the algorithm. For ML/AI functions, the degree of “garbage in” can affect how unpredictable, unexplainable and goal uncertain the algorithm performs (Amodei 2016).

Machine learning is a process where input data is used to train the algorithm to determine a correct answer. In general, training data sets have two parts: (1) the attributes that the function is learning to recognize, most times called instances, and (2) a truth label that describes the categorization of those attributes to train on correct answers. A trained ML function receives attributes and determines whether those attributes belong to a category such as a dog or cat. This research investigated measuring various aspects of attributes used for categorization. In the research, we divided attributes within the training set into three levels of significance: (1) primary, (2) secondary and (3) tertiary. In our sandbox analysis, we considered and determined that tertiary was unnecessary with regard to its modality. Our concern focused on the effects missing or sparse data occurrences had on the most significant attributes, i.e., a noisy operational environment where the unexpected happens. Unexpected examples might be communication link failures, sensor malfunctions or human data input error.

Training set size and composition (Foody 1995) is the principal ingredient that establishes the quality and quantity of a Machine Learned algorithm. No matter how exceptional is your Data Scientist development team, without the adequate quality and quantity of training data, the algorithm will never meet operational needs. The problem is that training data is a new paradigm for acquisition managers to consider. There are methods to test the output (Pei 2017) to determine incorrect corner case behaviors. Some of this tests are provide “whitebox” analysis. Yet, even these tests don’t provide insights into the composition and size of the input, again raising the concern about “garbage in, garbage out.” Key areas needing to be addressed regarding “negative training” (Rodríguez-Pérez 2017), i.e., things to not categorize, or how well noisy data occurs within a “realistic” operational environment cannot be addressed without looking at input. How well does input represent missing and sparse data issues and how much of the data set training consist of these examples. Testing the output or reviewing the array of weights inside the box might provide insights, but direct measurements of inputs will provide facts.

Instead of the output performance, our approach focused on measuring training data input as an approach to increase algorithm success rate reliability (Kim 2014). Preparing the training data for measurement is a form of curation within the Data Science field. This type of curation rigor of the input will aid the developer in thinking about how "noise" might affect the algorithm when deployed in its operational environment. Obviously, it is always important to measure output, but this research demonstrates the value of detailed measurements of the input.

Training ML algorithm based on operational environment "realism" was the primary motivation behind the development of these measurements. Since "realism" was the goal, it was necessary to create a program that represented products that would eventually be deployed in the operation environment. For the purpose, a "Sandbox" was created.

Our "Sandbox" – Because "Seeing is Believing"

Results of this research are based on using a "Sandbox" implementation approach that represents completion of three phases of a four-phase research approach. A "Sandbox" implementation approach means that a project was created, stakeholder requirements generated, architecture defined, design constructed, code developed and tests conducted within a confined, controllable environment for training, experimentation and analysis. Our Sandbox is designed to support seven different AI-enabled algorithms. To support realism, a mocked up acquisition program was created that consisted of five different AI-enabled algorithms supporting a mission planner and three different AI-enabled algorithms supporting two deployed autonomous vehicles. Both mission planner and autonomous vehicles had a full set of DoDAF system diagrams and UML Sequence Diagrams defining interfaces associated with software message transfer, SQL commands and application programming interfaces (APIs). These artifacts were designed in detail and reviewed before proceeding with ML algorithm investigation and development. Using this process allowed for an understanding of needed measurements regarding training sets. We also developed and reviewed a graphic user interface (GUI) and how human interaction plays a role in safe AI.

The Sandbox provided an opportunity for experimentation of an integrated hybrid system, combining various AI technologies to represent advanced capabilities (Baum 2011). This hybrid system allowed for "what-if" variations and intentional mistakes to investigate and test various measurements and approaches that could affect accurate forecasts and thereby resolve ML behavioral issues in advance. The following ML algorithms were either coded or design reviewed for implementation in the Sandbox: (1) for the Mission Planner -- Naïve Bayes, Logistic Regression, Random Forest, k Nearest Neighbor and XG-Boost, and (2) for the autonomous vehicles – Deep Neural Network, Deep Reinforced Learning and Convolutional Neural Networks. Algorithm design review included hyper-parameters variations specific to the algorithm under investigation.

The eventual goal of the "Sandbox" is to develop code and analytical measurements for all five different AI-enabled algorithms supporting the mission planner and all the three different AI-enables algorithms used within the autonomous vehicles. Within the sandbox environment, identified AI-enabled systems were analyzed and the measurements described below were identified to address the issue of how quality and quantity of training data might affect the confidence in the behavior of the algorithm in a deployed environment.

This phase of the research has resulted in postulating 14 tips that include best practices and measurements spanning requirements, architecture, design, development and test. All 14 tips focus on how to improve confidence in ML algorithm behavior. This paper presents results

associated with two key tips regarding measurements to determine if there is adequate quality and quantity of data within a training set for the ML algorithm to meet operational needs. The measurement approaches described in this paper are to demonstrate the reliability of the training set in establishing confidence in the behavior of the machine learned algorithm.

The paper will highlight insights into both the quality and quantity of the attributes within the training set instances. An instance is a single sample of data used for training the algorithm. The motivation of this research is to include the proposed measurements as part of Objective Quality Evidence (OQE) gathering when submitting recommendations by system safety practitioners for Weapons Systems Explosive Review Board (WSESRB) review in support of justifying ML behavior confidence. In addition to OQE, the research findings will also provide valuable insight to the acquisition community, to include program managers, and test and evaluation engineers.

Training Data Modality

When creating training data, it is important to understand the operational environment being represented in order to ensure adequate development of the ML algorithms. The training data is either found from live events or synthetically created to match the operational scenario that will be provided as input to the ML algorithm. Therefore, the ML algorithm must learn how to perform under these conditions. Three types of modality represent various operational environments that can be encountered during deployment, where the type of modality defines how the ML algorithm needs to be trained.

ML Training Data Modality 1: This modality supports training data sets that are based on an operational environment from multiple data sources, where each source contains one or more attributes as described in Figure 1. In Figure 1, the various sources of separate data attributes is either found from live events or synthetically simulations created to match the deployed operational scenario. Therefore, the input for ML algorithm for training needs to replicate the input that will be received during deployment.

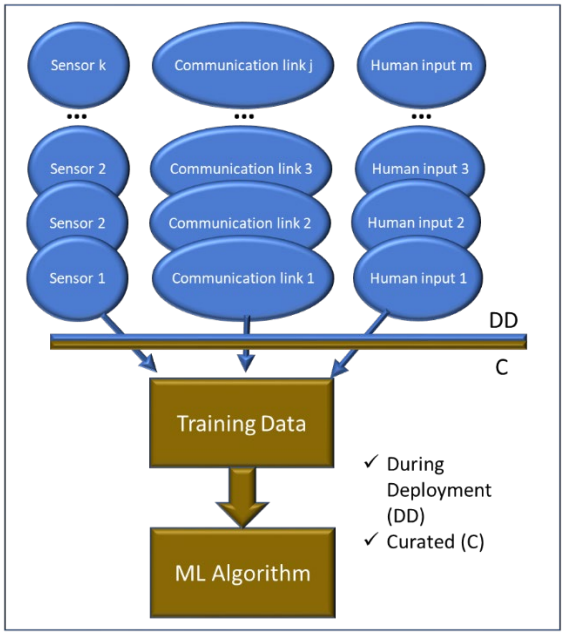

Figure 1. ML Training Data Modality 1

ML Training Data Modality 2: Training data sets that are based on an operational environment from a single data source, where the single data source contains multiple data attributes as described in Figure 2. In Figure 2, the one stream set of aggregated attributes is either found from live events or synthetically simulations created to match the deployed operational scenario. Therefore, the input for ML algorithm for training needs to replicate the input during deployment.

Figure 2 represents several versions of Modality 2, labeled (a), (b), (c) and (d). Version (a) describes the simple case where a sensor is capturing an image (in some frequency spectrum) that contains all the attributes needed to train the ML algorithm. As in all versions, Version (a) contains all the attributes needed to train the ML algorithm based on how the algorithm will be operationally deployed. Version (b) describes how one sensor might create a string of images causing channels for the ML algorithm to learn. Each channel might require one algorithm or a unique set of algorithms for processing. Version (c) describes a series of images, similar to Version (b), but in this case, as in an attempt to capture a 3-D image, where the combination of each slice of the image may constitute a single attribute that is part of the training. Finally, Version (d) describes how multiple attributes sources might be fused/combined into one source that will be used for training the ML algorithm. Again, the selection of Modality 2 Versions is based on the operational need/requirements associated with its deployment.

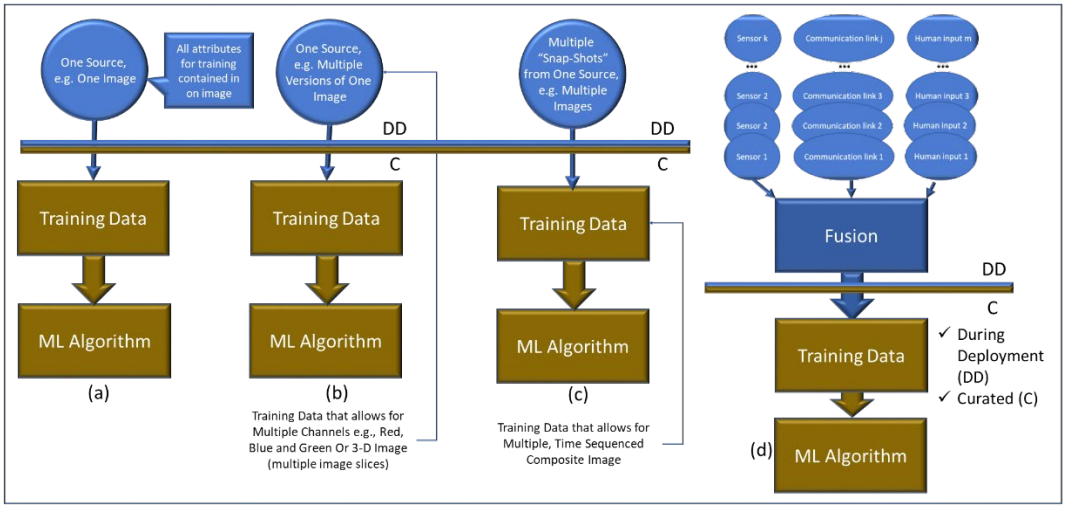

Figure 2. ML Training Data Modality 2

ML Training Data Modality 3: Training data sets that are based on an operational environment from a combination of multiple data sources where each source contains one or more attributes form various sources and from a single source containing multiple aggregated data attributes.

Modality 3 is the most challenging data set to replicate or find that can adequately represent "realistic" operational environments. In all three modalities, the primary challenge with using adequate data sets for ML training is to ensure the training set accurately represents "realistic" operational environments. The more complex in the composition of data sources that the ML algorithm needs for training in order to adequately perform its function, the more challenging it is to replicate a "realistic" training set that includes issues, such as communication failures over data links, unintentional human input error or sensor malfunctions. Additional challenges stem from adequately replicating noise that blur, surround or somehow challenges the data source feeding the ML algorithm. For example, synthetic replication of a single attribute

over various slices of an image may be difficult to create with the adequate noise background, e.g., the blur needs to be consistent. The difficulty increases when that attribute needs to train the ML algorithm using hundreds of slight variations. Complications increase when dozens of attributes need to be included within the slices of images that will constitute a training set will realistically represent the required operational environment.

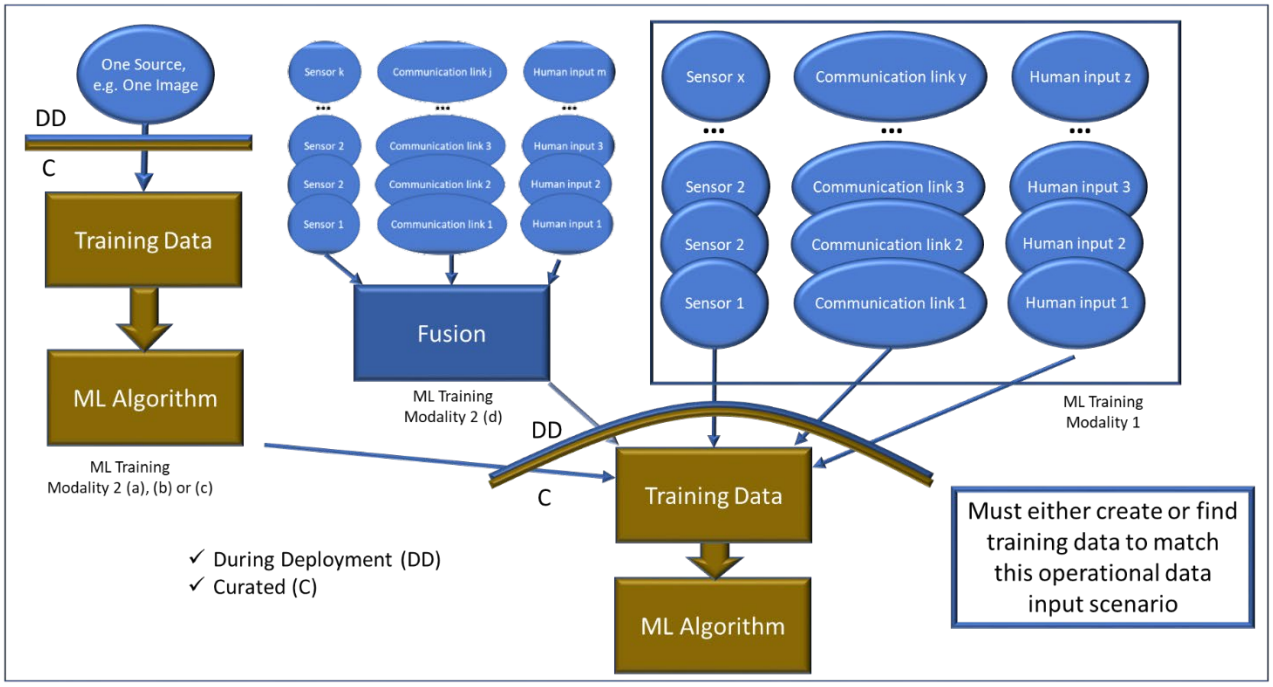

Figure 3. ML Training Data Modality 3

Given these potential challenges, a need to measure the quality and quantity of the data set to meet the operational needs of a "realistic," noisy environment become essential to ensure confidence in the behavior of the ML algorithm during deployment.

Missing and Sparse Data Effects on Modality

For Modality 1, missing data can be represented as a sensor, communication link or human input issue. The sandbox software filtered the faulty data as being out of performance bounds and therefore provided no values. Sparse data occurred when the sensor, communication link or human input was properly working but data was unavailable for the ML algorithm to use. In this case, the sandbox implementation design provided zero values for those data sources. The sandbox software handled the zero values as no input to the ML algorithm. The challenge for Modality 1 is that missing and sparse issues can occur at the same time from different data sources. Either the data source fails, causing missing data or the data source does not register any input. In either case, the sandbox software filtered the sparse or missing data and therefore provided no values in those situations. Therefore, when developing AI functions, the training data needs to represent these occurrences and the developer needs an approach to handle the occurrences of sparse or missing attribute data. In the case of the sandbox, the training set consisted of secondary attributes when the primary attributes were not present. Primary attributes represent nominal input expectations, whereas secondary attributes are a back up to the unexpected.

For Modality 2, we still used our previous definition of missing or sparse data, when either occurs from a single sensor. Because a single source was replicated, only one could occur but not both. From our sources, this was an accurate representation of a realistic operational environment. For Modality 3, the combination of missing and sparse data could occur causing significant replication issues with regard to the training data set supporting a "realistic" operational environment. It is important to note that in Modality 2 and 3, missing and sparse data issues become even more challenging because filter techniques, like used in our sandbox, are harder to apply. For example, if missing data occurs, then the sensor may be malfunctioning causing blurs in the image, which would require the developer to train on secondary attributes that compensate for this type of blur in the picture. Once the first algorithm failed to categorize above threshold, this compensation approach may require a second algorithm trained on primary and secondary combination of attributes within the image. If the image contains no attributes, potentially from a sparse data issue, then categorization is impossible, there would be no secondary attributes to use. Again, complexity of how the training data is composed becomes more challenging, but needs to be addressed as part of oversight.

Training Set Composition Measurements

In developing our measurement approach to better ensure a training set represented a "realistic" operational environment, it was important to measure how well the training data represented both in quality and quantity missing and sparse issues with the data sources. Table 1, rows (a), (b), (c) and (d), represents questions that need to be addressed based on Modality, first by adequately defining the operation environment the training set represents based on modularity and then by ensuring the quality and quantity of data is adequate for the ML algorithm training.

	Investigation Topic	(Modality 1) multiple data sources, where each source contains one or more attributes	(Modality 2) single data source containing multiple data attributes, e.g., CNN	(Modality 3) combination of multiple data streams, where each stream contains one or more attributes and from a single data stream containing multiple aggregated data attributes, e.g., Naïve Bayes aggregated with CNN
(a)	Data Source Precedent for Improving Success Rate (ranking of primary, secondary tertiary… n attributes)	Which sensor, communication link or human input content elements take precedent over others for improving success rate when training the ML algorithm under normal to stressed operational conditions?	Which attributes within the single data source take precedent over others for improving success rate when training the ML algorithm under normal to stressed operational conditions?	What data source content is more significant with regard to normal to stressed operational conditions? When dealing with separate streams, which sensor, communication link or human input content elements take precedent for improving success rate when training the ML algorithm under normal to stressed operational conditions? When dealing with combined streams, which attributes within the single data source are identified as primary, secondary and tertiary regarding importance for ML algorithm to improve success rate under normal to stressed operational conditions?
(b)	Missing and sparse data issues modeled	How is sensor malfunction, message corruption and human input errors on the higher precedent attributes forcing lower level attribute mixes of training data to ensure algorithm can deal with "real" operational issues?	Corruption in parts of image, especially containing higher precedent attributes forcing secondary and tertiary attribute mixes of training data to ensure algorithm can deal with "real" operational issues.	Combinations on modalities 1 and 2 regarding training of algorithm to deal with "real" operational issues.
(c)	Quality of Training Data Characterized	What is the precedent list (from highest to lowest) of attributes being used for training.	Same as Modality 1 for this row.	Same as Modality 1 for this row.
(d)	Quantity of Training Data Characterized	How much more emphasis Is placed on quantify of training data variations that have higher precedent than lower?	Same as Modality 1 for this row.	Same as Modality 1 for this row.

Table 1. ML Training Data Investigation Topics by Modality Types

In Table 1, each row represents a series of questions associated with the modality of the ML Training set. Row (a) introduces the need to group attributes in terms of precedence/significance with regard to an expected operational norm and potential source failures (causing Missing and Sparse data issues) that the ML algorithm needs to learn in terms of data inputs. What attributes are primary to consider for training the ML algorithm? What attributes are secondary? Depending on the operational environment, there may be "n" number of groupings. Row (b) focuses on missing and sparse data modeling of attributes for training. As

described in the previous section, missing and sparse data issues can direct attribute precedence. For example, when one primary attribute is not available, can another attribute in the secondary group be used to increase behavior confidence? Is the ML Algorithm being trained to use primary and secondary combinations of attributes? Row (c) defines an approach to analyze quality based on a precedence list. Within each group, what is of highest precedence for the ML algorithm so it can be adequately trained? The answer to this question ensures that the developer understands the relationship between attributes and the operational environment those attributes will support. Finally, Row (d) focuses on the need to understand if the quantity of training data is sufficient. Although quantity may be analyzed using overfitting and underfitting techniques specific to the algorithm being trained, this quantity analysis is based on how much more emphasis is placed on training data with higher precedence vs lower precedence. For example, if higher precedence/significant attributes are based on nominal operational conditions, then by definition of precedence/significance, slight variations of higher precedence attributes should have a greater or at least equal number of instances as compared to lower precedence attributes. If this is not the case, then why is one attribute group more significant over the other? Row (d) topic of investigation asks the questions, “Is there sufficient training data based on precedence grouping?” Measurements described in this paper provide answers to the topic investigation questions shown in Table 1.

Training Data Measurements

Using the sandbox, we created and examined two types of measurements that support answers to the questions posed in Table 1. The focus of both measurements is on attributes/features within each sample/instance.

If synthetic data is created, then a Design of Experiments (DOE) review needs to be performed during the requirements and architecture stages, e.g., somewhere during preliminary design review (PDR) and critical design review (CDR) timeframes. This become obvious while working within our sandbox development environment. In our sandbox, we used a modeling and simulation (M&S) approach to create training data. We developed a DOE that ensured primary and secondary data sources were created to support five ML classes using various combinations of seventeen attributes. Figure 4 represents the sandbox data sources, real-time and synthetic, and includes the operational environmental variables that are translated into 17 attributes supporting five classes/categories.

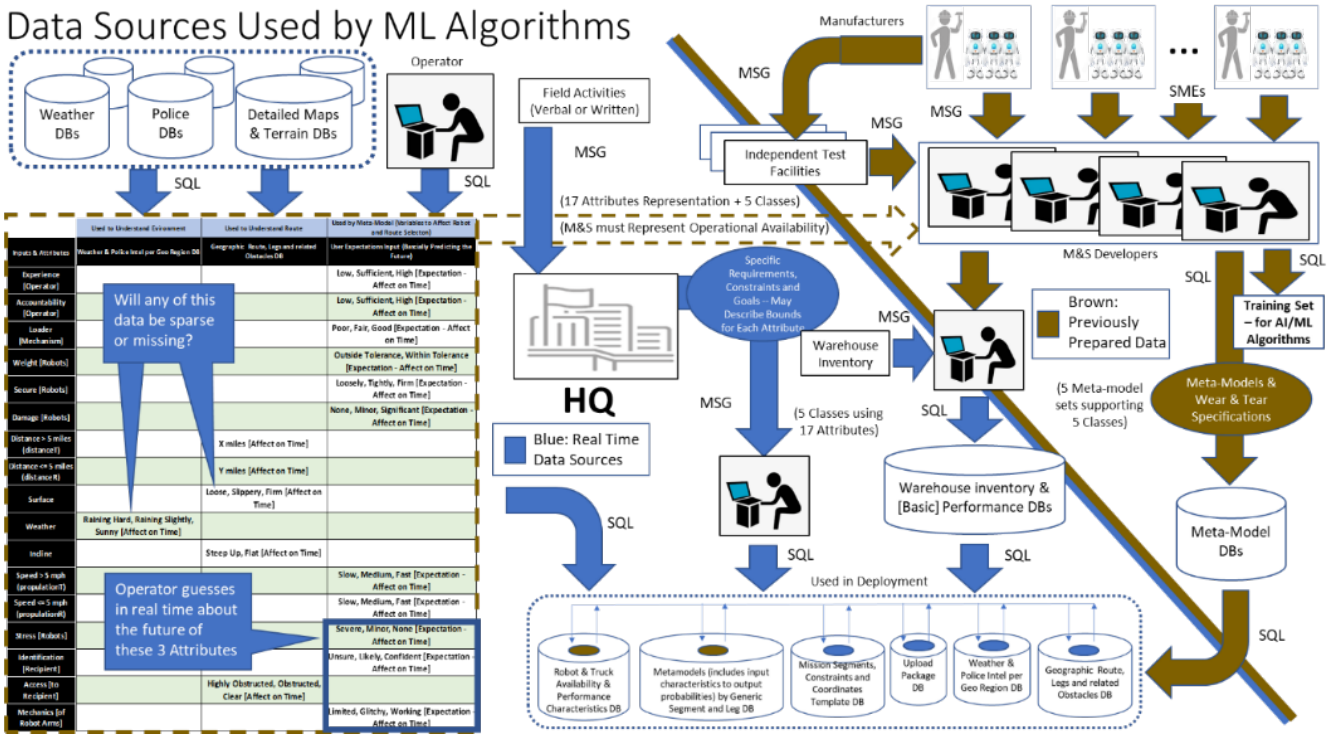

Figure 4. Operational Environment being Represented using M&S Synthetic Data

Training Set Alignment Test (TSAT)

Training Set Alignment Test (TSAT) is an approach we recommend using during requirements definition, architecture review and finally during algorithm code analysis. When creating synthetic training set data via modeling and simulations, attributes from highest to lowest significance should be identified in the Design of Experiments (DOE) to ensure proper emphasis is placed on primary, secondary to n-levels of precedence.

In our sandbox analysis using synthetic training set creation from simulations, we were able to develop the TSAT measurement process. Figure 5 diagrams the steps discussed below when taking a TSAT measurement:

- At Requirements stage and checked during Architecture review
    - First Step: Determine what attributes are most significant as compared to others in terms of the function the ML algorithm must perform. Note: this is based on the part the algorithm plays in the mission. What functions must it perform so the other subsystems can achieve their goals? For example, a common ML algorithm function is computer vision. What are the most significant attributes it should use to perform its image recognition function?
    - Second Step: Group the most significant attributes and consider them as primary attributes to the algorithm's learning process
    - Third Step: Group the other algorithms in terms of secondary and tertiary significance in terms of what the algorithm needs to learn

- When training set is produced, conduct analysis (Note: we've included the next three steps as part of Algorithm Code review because training data creates the weights and structures that constitute the deployed code).
  - Fourth Step: When the training data set is generated/gathered, use the statistics of how often an attribute occurred to determine the ranking.
  - Fifth Step: Perform a weighted calculation (similar to a discrete match filter in signal processing.)
  - Sixth Step: Determine if this grade, meaning the determination of how well the DOE goal matches the generated/created data. In this approach, the grades range up to 100%, where 100%i s a perfect alignment between operational needs and training data, where below 25% is extremely poor. Even with the most tolerant requirements, it is recommended that anything below fifty percent should not be accepted.

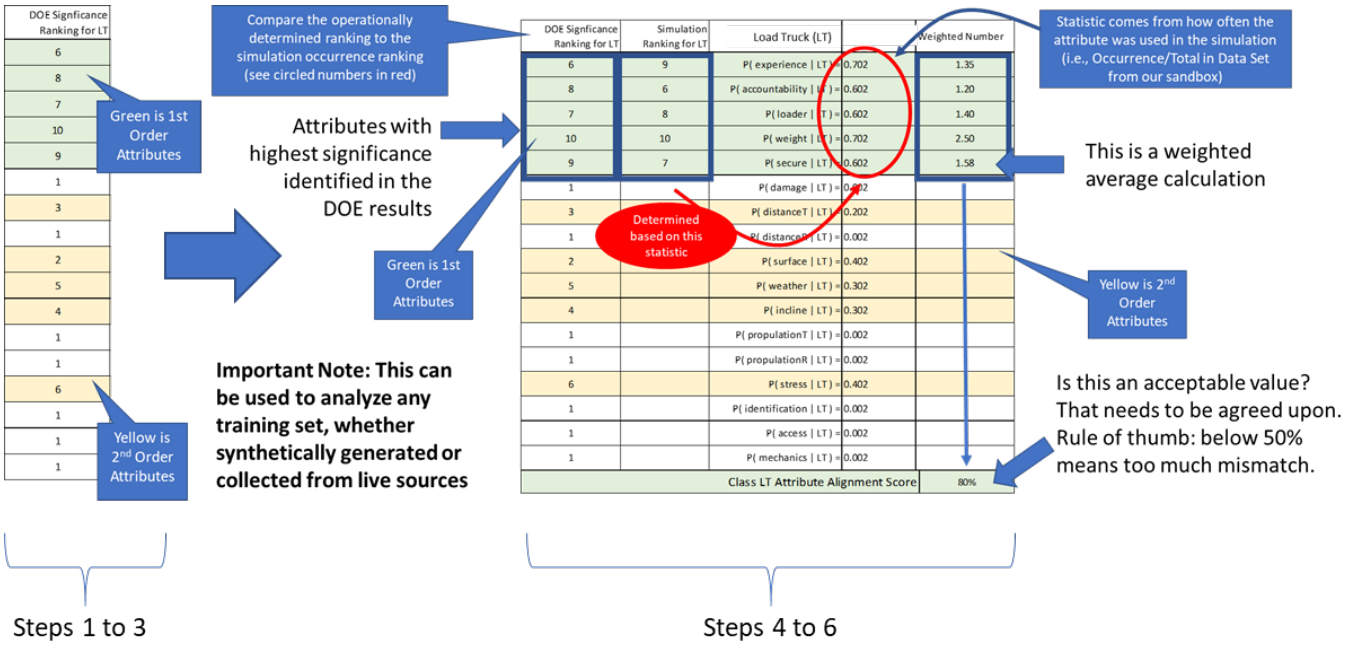

Figure 5. TSAT Diagram of Steps

TSAT ensures that the developer verifies that the attribute priority and ML training set modality is congruent between the deployed architecture and the training data set generation process based on the precedence/rating of attributes defined in TSAT. This will also document compliance to requirements for review.

Procedure for calculation:

1. Determine a scale for grading from 1 to "m," where "m" means greatest attribute priority/significance based on operational deployed needs.
2. Identify attributes a_1 to a_n to grade, such that "n" is the number of attributes being graded out of r total attributes available. Therefore $n \leq r$ and $n \leq m$, where grading a_i with grade "m" indicates a_i (m) is the most important attribute based on operational needs. Additionally, attribute grading range is (m-n+1) to m, consecutively, where lowest grade indicates least operationally important (possibly DOE analysis and/or SME determination).

3. Identify the n attributes that occur the most times in the training data. Using the same scale "m," grade attributes b_1 to b_n based which attribute occurred the most often within the training set (this can be a statistical number, e.g., 70% of the time b_i attribute was used in simulations or 70% of the samples/instances were collected, e.g., images, that contained attribute b_i). Again, grade "m" indicates b_i occurred the most and (m-n+1) indicates b_j occurred the least within the training set.
4. Perform $k =)$ and $\beta =* bi(grade) \leq m$
5. Perform $* 100 = \alpha\% \geq 50\%$ as a constraint

Source to Attribute Ratios – nth Order Grouping (StAR-n)

Source to Attribute Ratios – nth Order Grouping (StAR-n) is an approach that can be used during requirements definition, architecture reviews and finally during code analysis. The basic premise is that attributes (e.g., primary, secondary or tertiary groupings) with the highest significance (precedence/rating) identified in the DOE (defined in TSAT) should be occur in greater numbers of instances within the Training Set than lower significance attributes. The comparison of numbers can be analyzed as ratios.

The reason why developers should verify that primary instances have greater numbers than secondary, and so on, is because: (1) With live data collection, there is a difficulty with finding or creating realistic training data that includes noisy environments representing missing and sparse data issues; and (2) With synthetic data creation, there is a physical limitation with how much simulation can be performed within the timeframe allotted? (Remember that most likely there is an infinite number of possibilities in terms of training data variations.) What should be the priority in your DOE?

In our sandbox analysis, we were able to develop the StAR-n measurement process. Figure 6 diagrams the steps discussed below when taking a StAR-n measurement:

- At Requirements stage and checked during Architecture review:
  - First Step: Create a ten by ten matrix, labeling each axis from zero to 1.
  - Second Step: Label the horizontal axis "% Number of Primary Attributes vs Total Attributes for Class" and the vertical axis "% Number of Primary Attribute Instances vs All Instances for Class"
  - Third Step: Determine a three-color zone scheme (see Figure 6 as an example), where green indicates that the ratio fell within acceptable limits, yellow indicates ratio is boarder line acceptable, and red color zone indicated ration is outside expected limits. Color of the zone should how well training data reflects operational environment. Based on color zone, determine evidence justification. Examples (used for guidance only) are described below:
    - Zone Green: Evidence of data by showing appropriate n-th order groups of training sets collected or generated by the simulations, including success rates as well as the TSAT results.
    - Zone Yellow: Zone Green evidence plus justification on why n-th group precedence can still handle the unexpected and provide acceptable success rates.
    - Zone Red: Zone Green and Yellow evidence as to how this algorithm is going to be supervised or monitored when operationally unexpected events occur.

- When training set is produced during Algorithm code review:

- Fourth Step: Calculate the σ and δ (see Figure 6 as an example) ratios. Each ratio should be less than 1. The example below is for primary attributes, but can be done for any n-th order attributes:
    - σ (by Class) = (Number of Primary Attributes / Number of All Attributes) ≤ 1.
    - δ (by Class) = (Number of all Primary Instances / Number of All Instances) ≤ 1.
- Fifth Step: Plot (x, y) using (σ, δ) pair of numbers and assess where the pair fall within the color zones to determine support action. An example is provided in Figure 6.
    - Zone Green: Evidence of data by showing appropriate n-th order groups of training sets collected or generated by the simulations, including success rates as well as the TSAT results.
    - Zone Yellow: Zone Green evidence plus justification on why n-th group precedence can still handle the unexpected and provide acceptable success rates.
    - Zone Red: Zone Green and Yellow evidence as to how this algorithm is going to be supervised or monitored when operationally unexpected events occur.

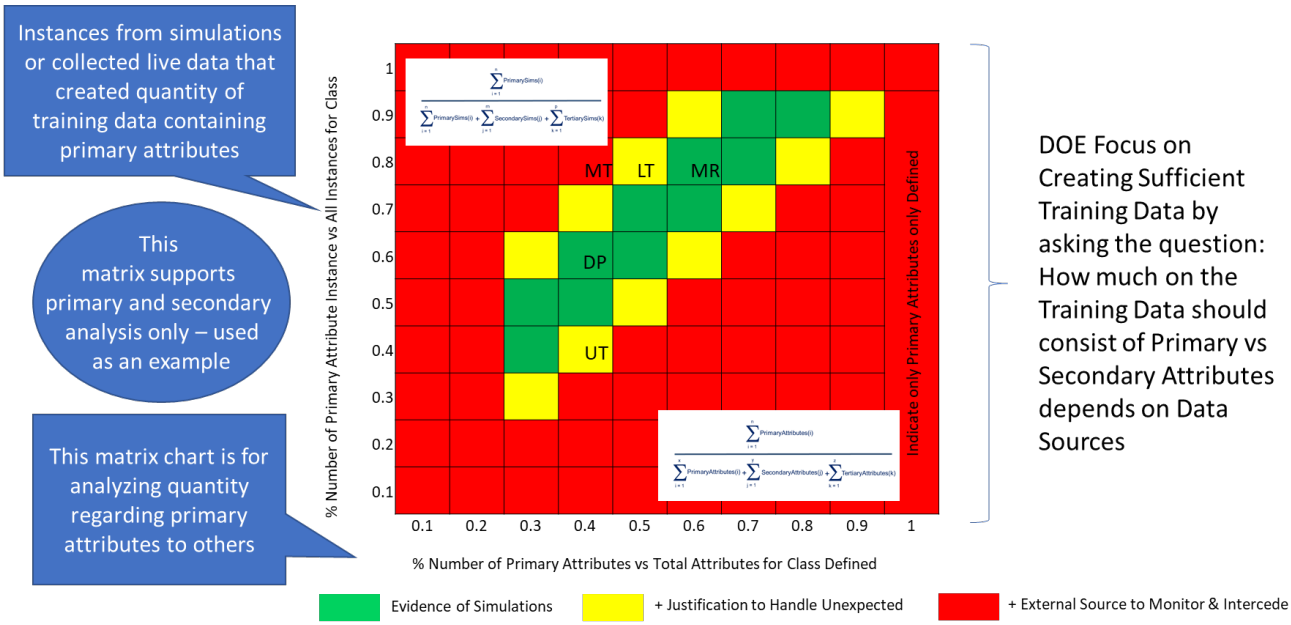

Figure 6. StAR-n Diagram of Steps

Notice that in Figure 6, the five classes have been plotted based on our sandbox results, provided as an example. Given the plot, there are two classes in the green, two in the yellow and one in the red. The color scheme relates to the type of justification needed in support of using the training data for those classes to adequately develop the algorithm to support the required operational environment. Since there is a significant mismatch when a (σ, δ) pair a plotted in the red zone, as in our example in Figure 6, we need to make sure that this algorithm has supervision when the expected occurs. Justification and the relationship to which boxes are colored needs to be described during requirements and checked during the architecture review. Again, the periods can be around PDR and CDR.

Matrices can be created for Primary, Secondary and Tertiary attributes, not just Primary. The StAR-n Grouping Matrix for this sandbox was only 2nd order. A StAR-3 looks at ratios of primary, secondary and tertiary attributes, as they are defined through requirements. As stated, training data is key to the development and the question becomes how much of the training data consist of primary vs secondary vs tertiary attributes as dependent on data sources that will be available in the field. Again, the issue becomes missing and sparse data during deployed operations.

StAR-n provides confidence to the system safety practitioner or test and evaluation engineer when the training data is generated synthetically and an attribute random generator is used. StAR-n ensures that justification is provided as Level of Rigor evidence based on primary, secondary, … n-th order attribute ratios to training data content ratios as part of the assessment of operational needs compared to what the training data contains.

In the Sandbox, an attribute random generator was used to create 15,000 simulations supporting 5 classes and 17 attributes. The analysis focused on determining the "Simulation to Attribute Ratios" for 1, 2 or 3 (nth) Order and graphed in a matrix to determine what type of rigor is needed to justify the ratio involved with each class being modeled via selected attributes. Consideration included how the attribute random generator creating the training data simulated an operational environment of sparse and missing data for the targeted algorithm to learn. The matrix using StAR-n identifies the need for the three types of justification, Zone Green, Yellow and Red, as described above.

StAR-n measures data source requirements, architecture and data set generation process specific to the categorized ratios of attributes defined. This measurement helps ensure that the developer is reflecting reality during algorithm development.

Combining TSAT and StAR-n ensures congruency between the operational environment and the training data set generation process. Figure 7 graphically describes the congruency using our sandbox classes between the blue operational deployment of the algorithms and the brown development of training data incorporating the attributes that will be available to the algorithm during operations.

It should be noted that labeling attributes/features, especially when live data is being collected, can be challenging. Labeling each instance within a Modularity 2 training set means looking at each sample and ranking, grouping and counting various n-order attributes. The challenge increases when evaluating effective sample size and the correlation between the attributes. Effective sample size affects the total number of instances used for training and therefore the classifier's performance/confidence interval (Figueroa 2012). Feature correlation affects total number of attributes used for training. Correlation can be measured and observed. For example, a smile affects two sides of the mouth. It would be inappropriate to consider both ends of the mouth as two different attributes. They are not independent events. Statistical independence of attribute within an instance affects algorithm training. A smile, a single attribute, can have many variations that affect both sizes of the mouth. Effective sample size is related to the randomness of each created or observed instance. Again, the samples should support statistical independence.

Instances and features within the instances need scrutiny to know if an algorithm is taught properly. Without this focus, the training would be uncontrolled. It would be like instructing children math and not knowing if they are being taught the "right" mix of problems or just the same problem with different letters for the variables. In training algorithms, it is important to ensure the training content is specific to the feature level of the education. This rigor is

practiced in life sciences (Toloşi 2011) where wrong conclusions might lead to fatality. This same level of rigor should apply to any function performing operations that could cause lives to be at risk. Therefore, as applied to DoD, effective sample size and attribute/feature correlation of each instance needs to both be assessed as statistically independent (or at an acceptable low correlation) when applying TSAT and StAR-n measurements to training sets of algorithms performing operations that could cause lives to be at risk.

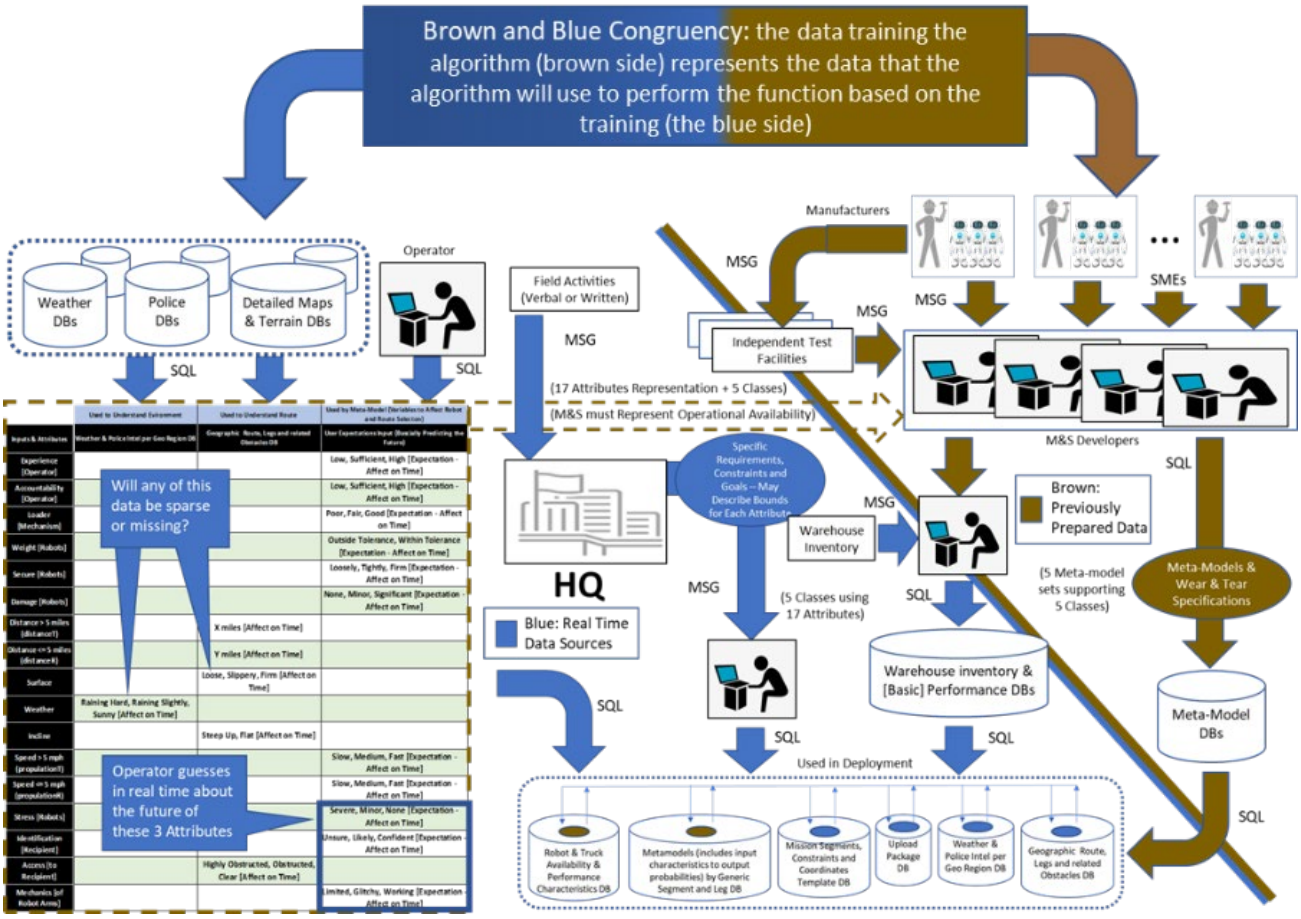

Figure 7. TSAT and StAR-n ensure congruency between what will be operationally deployed and what will be synthetically developed or collected from live data sources.

Numerical and Graphical Interpretation of Measurements

Using sandbox generated training sets, both TSAT and StAR-n were applied. Figure 8 and Table 2 represent the TSAT analysis for 17 attributes used by five classes from our sandbox.

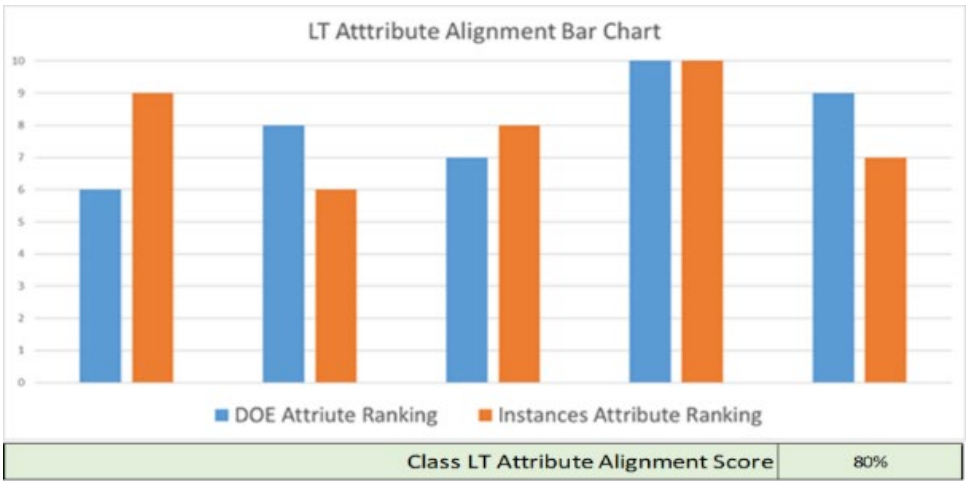

Figure 8. Sandbox Results from Applying TSAT Measurements for the LT class.

In Figure 8 (one of the five classes), you can notice that the blue and red bars are fairly equal in height causing a high score of 80%. Visually, if blue and red bars have significantly different heights, then a lower score will occur.

Class LT Attribute Alignment Score			80%	
		Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance		3.518333333	70%	50%
Secondary Data Source (2nd order significance		1.485	30%	50%
Tertiary Data Source (3rd order significance		0.002333333	0%	0%
	TOTAL	5.005666667	100%	100%

Class MT Attribute Alignment Score			87%	
		Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance		4.085333333	68%	46%
Secondary Data Source (2nd order significance		1.919	32%	54%
Tertiary Data Source (3rd order significance		0.001333333	0%	0%
	TOTAL	6.005666667	100%	100%

Class UT Attribute Alignment Score			79%	
		Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance		0.785	32%	45%
Secondary Data Source (2nd order significance		1.685333333	68%	55%
Tertiary Data Source (3rd order significance		0.001666667	0%	0%
	TOTAL	2.472	100%	100%

Class MR Attribute Alignment Score			88%	
		Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance		4.468666667	74%	60%
Secondary Data Source (2nd order significance		1.534666667	26%	40%
Tertiary Data Source (3rd order significance		0.002333333	0%	0%
	TOTAL	6.005666667	100%	100%

Class DP Attribute Alignment Score			85%	
		Probability (Weight) Totals	Sim Ratio	Attribute Ratio
Primary Data Source (1st order significance		4.61866667	41%	60%
Secondary Data Source (2nd order significance		6.679077963	59%	40%
Tertiary Data Source (3rd order significance		0.002333333	0%	0%
	TOTAL	11.30007796	100%	100%

Table 2. Attribute Alignment Scores for Each of the Five Classes

When looking a StAR-n ratios, Figure 9 describes a visual inspection of the two axis in the matrix. The “Instances Ratio” represents the vertical axis, “% Number of Primary Attribute Sims vs All Sims for Class.” The “Attribute Ratio” represents the horizontal axis, “% Number of Primary Attributes vs Total Attributes for Class.” In Figure 9, there are an equal number of primary and secondary attributes, as seen in the “Attribute Ratio” graph. In the “Instances Ratio” graph, although equal in number, there are more primary attributes in the training set.

Figure 9. Visual Review of Instances Ratio to Attribute Ratio in StAR-n Matrix

To better understand the significance of the ratios, consider the combinations of training instances based on use of the sandbox. In Figure 10, just focusing on whether an attribute, i.e., data source, will be present or not in the operational environment. Class combinations range from 821 to 2026, totaling 9580 different combinations. If we decided to train our ML algorithm to recognize a class based on attribute presences and numerical integer value, the combinations become extremely large. If we decide to have values in the real number domain, the combinations become infinite.

How much training data can you generate or collect to support 9580 combinations, in the simple case, or an infinite number of values in the extreme case? Therefore, it is necessary to prioritize attributes in terms of the number and type of instances within the training set. That is why TSAT and StAR-n are vital measurements.

Design of Experiments	1st Order Significant Attributes	2nd Order Significant Attributes	Total 1st and 2nd Order	Keep 2 1st Order	Keep 3 1st Order	Keep 4 1st Order	Keep 5 1st Order	Keep 6 1st Order	Total Configurations
LT	5	5	10	560	210	30	1	0	821
MT	6	5	11	1890	1120	315	36	1	3384
UT	*5*	*6*	11	840	280	35	1	0	1178
MR	6	4	10	1050	700	225	30	1	2026
DP	6	4	10	1050	700	225	30	1	2026
									9580

Figure 10. Attribute Occurrence Combinatorial Variations of Primary and Secondary Attribute Types per Class within Sandbox

From this research, three guidelines when using StAR-n to analyze ratios consistently surfaced.

Guideline 1: Order of precedence/significance should also describe ratio structure. Primary should have more instances than Secondary, Secondary should have more instances than Tertiary, etc. Describing the obvious, you would not want the ratios to be inverted, meaning the secondary would have more secondary attributes than the primary.

Guideline 2: Depending on the n-th order grouping of significance, there should be instances, therefore ratio values, for all n-th order combinations. Figure 11 describes when a tertiary attribute group, equal in number to the primary and secondary attributes, was omitted from the modeling and simulation. This means that the training data will not support the operational need associated with its deployment.

Guideline 3: If one of the n-th order grouping attributes is less than 5% (conservatively) in the attribute ratio graph, consider including it in other attribute groupings. Remember that a lower order attribute is likely to have less simulated or live data instances collected. This means that the instances will be lower than 5%, and likely be between 1 and 2% if Rule 1 and 2 are followed. Therefore, it may make more sense to include this n-th order attribute into another attribute group.

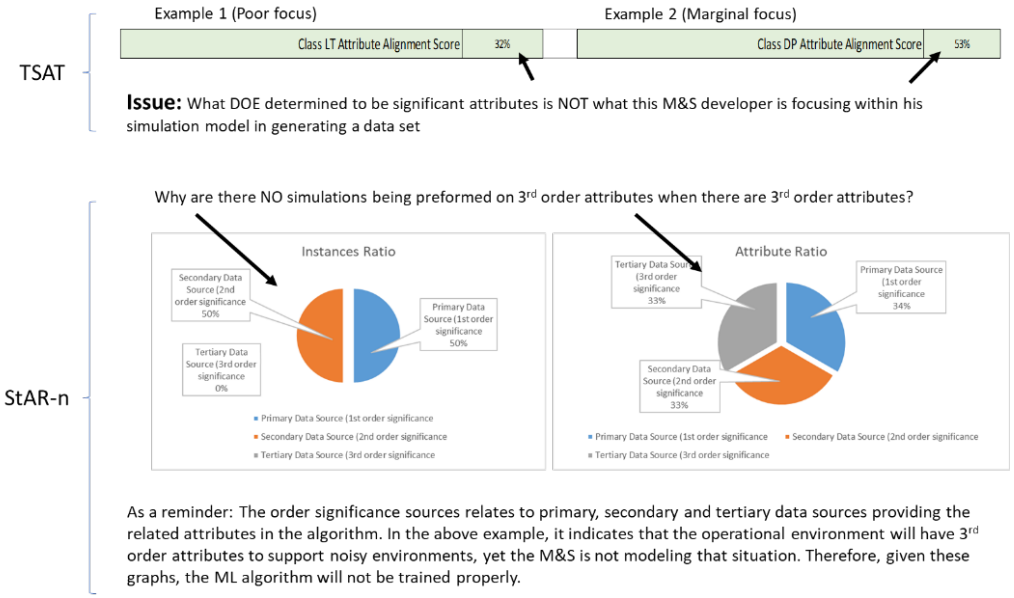

Figure 11. TSAT and StAR-n Identified Issues.

By following the three guidelines, valuable discussions can occur with the developer. As a reminder, the order significance sources relates to primary, secondary and tertiary data sources providing the related attributes in the algorithm. In Figure 11, tertiary data source creating attributes is not included, like our sandbox example, which means tertiary attributes i "not significant" from DOE viewpoint. It could be that there is less than 5%, which is causing the tertiary attributes to not be considered for ML algorithm training. In our sandbox, there were no tertiary attributes.

The Figure 11 example provides discussion points as follows:

- In Attributes Ratio, notice that 1st and 2nd attribute grouping have about the same number of attributes. Yet, if primary and secondary attributes are equally produced in the simulation, why is one group considered more significant than the other? This may not be wrong but definitely a discussion point for the developer. In this example, maybe 2nd order plays important role acting as a noisy environment or non-ideal environment. If of equal importance, then maybe there are only primary attributes. If so, is a noisy environment still being modeled? No right or wrong answers, just discussions that should be had on the operational environment and how the ML algorithm is being trained.

- In this example, there is a potential issue: If 1st and 2nd order are about the same number of simulations and this needs to be understood. What also needs to be discussed is why the fraction of attribute occurrence in the training set is so disproportionate to the fraction of total attributes? (Again, 2nd order acts as noisy or none ideal environment)

- Should the 2nd order and 1st order be a different ratio given the attribute numbers are about equal?
- Consider a better ratio, possibly 33 % to 66%, meaning I've run twice as many simulations on the 1st order vs the 2nd order.

Findings

From a sandbox approach, Table 3 became evident in terms of how much AI and traditional software code differ with respect to the acquisition process:

Category	Traditional Software (designed to provide a "certain" outcome)	ML/AI "Likelihood" Software (designed to provide a "likely" outcome)
Architecture	Architecture can vary.	Architecture needs to be portable and modular, e.g., micro-service and DEVOPS**.
Design	Design focuses on the actual development of the deployed code – the code is designed to provide a certain, specific outcome	Design focuses on the training data development that will create the deployed code – the code is designed to provide a likely outcome
Development	Development of code is done directly – coder determines the logic and math to use and codes it	Development of code is done indirectly – coder determines the logic and math used in the training (machine learning techniques), which results in the code
Test	Test is done on the code directly developed by the developer and to be deployed to prove certainty of an outcome	Test is done on the code (see below) developed from the math and manipulation of training data, i.e. use of machine learning techniques to prove likelihood of an outcome. If limited truth data, only use for testing.
Code Debugging	Can debug code that will be deployed	Cannot debug code that will be deployed – must retrain to create new weights and/or statistics for deployment
Transparency	The logic and math of the developer is shown directly in the code	The logic and math of the developer is not shown directly in the code
Critical Function Analysis	If a function is determined critical, the developer can have the math and logic challenged – focusing on the deployed code	If a function is determined critical, the developer can defend how the training data was created and manipulated – focusing away from the deployed code

Table 3. Traditional Logic vs Likelihood Software– Why treat ML/AI Algorithms differently in Acquisition.

A key question asked in this paper was, "Can the safety question with regard to weapon deployment regarding autonomy/AI ever be answered?" This paper answers this question in terms of rigor with regard to the training data. The measurements focused on improving confidence to an acceptable standard defined in requirements, checked during architecture and validated when reviewing the algorithm coding practices. To improve confidence of ML/AI behavior within the sandbox, TSAT and StAR-n measurements focus on n-th order grouping of attributes based on nominal operations for primary grouping, and non-nominal operations for lower level grouping. The cause of non-nominal operations is noise or faults in the deployed system. As described in previous sections, noise or faults result in missing and sparse data. How missing and sparse data affect the training data is based on the type of modality, as was discussed. TSAT and StAR-n measurements allows for ML algorithm training that ensures a match between the training data set and reality in the operational environment.

In the Sandbox, an attribute random generator was used to create 15,000 simulations supporting 5 classes and 17 attributes. The analysis focused on determining the "Simulation to Attribute Ratios" for 2nd Order analysis and graphed in a matrix to determine what type of OQE rigor was needed to justify the ratio involved with each class being modeled via selected attributes. Consideration included how the attribute random generator creating the training data simulated an operational environment of sparse and missing data for the targeted algorithm to learn. The matrix, using StAR-n, identifies the need for various types of rigor described previously based on where it is located in the matrix.

TSAT and StAR-n demonstrated that these measurement can support quality and quantity factual analysis that can be used by the acquisition community, including system safety and test and evaluation groups, to improve the confidence of the behavior of the algorithm to support a realistic deployment operations. From these measurement processes, issues associated with training, i.e., "garbage in," can be identified and resolved in advance and thereby increase ML functional confidence.

In our analysis, we were able to successfully use TSAT to ensure synthetic data for each of the five classes and 17 attributes had adequate quality and quantity of training data. The basic premise is that attributes (primary, secondary and tertiary) with the highest significance identified in the Design of Experiments (DOE) should be simulated more than attributes with lower significance. TSAT can effectively analyze Primary, Secondary to an n-th order data sources to determine if the training data is adequately aligned the Operational Needs defined in the DOE (note that the DOE must match operational use cases associated with mission parameter and environment).

1) We were also able to successfully use StAR-n. The Star-n can effectively use ratios involving primary, secondary to an n-th order, as they are defined by requirements and described in the architecture. As stated, training data is key to the development and the question becomes how much of the training data consist of primary vs secondary vs tertiary attributes, etc., as dependent on data sources that will be available in the field. StAR-n provides confidence to the system safety practitioner or test and evaluation engineer whether the training data is generated synthetically or collected from live events. StAR-n ensures that a Level of Rigor is provided based on primary, secondary and tertiary attribute ratios being with expected values.

By using both StAR-n and TSAT, the sandbox proved that quality and quantity of training data can be assessed. For TSAT, quality assessment meant the correct ranking of attributes (including primary, secondary, etc. mixes) that represented real world deployment issues associated with data source availability, which included noise factors. For StAR-n, quantity assessment meant the appropriate amount of samples/instances used for training based on operational priorities, which considered the mix of n-th order attribute ratios. Combining both measurements provides the Acquisition community, from project managers to test and evaluation engineers, the ability to maintain positive control over knowing that an AI/ML algorithm have been rigorously developed to support the expected behavior during deployment, even worst case environments. Why? Because using these measurements ensured that those environments were captured using the adequate quality and quantity of samples/instances needed to train the ML algorithm to deal with those issues.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P. F., Schulman, J., & Mané, D. (2016). *Concrete problems in AI safety*. arXiv. https://arxiv.org/abs/1606.06565

Armstrong, S., Bostrom, N., & Shulman, C. (2016). Racing to the precipice: A model of artificial intelligence development. *AI & Soc*, *31*, 201–206. https://doi.org/10.1007/s00146-015-0590-y

Baum, S. (2017). On the promotion of safe and socially beneficial artificial intelligence. *AI & Soc*, *32*, 543–551. https://doi.org/10.1007/s00146-016-0677-0

Baum, S. D., Goertzel, B., & Goertzel, T. G. (2011). How long until human-level AI? Results from an expert assessment. *Technol Forecast Soc Change*, *78*(1), 185–195.

Bohannon, J. (2015). Fears of an AI pioneer. *Science*, *349*(6245), 252.

Borrie, J. (2014). Humanitarian reframing of nuclear weapons and the logic of a ban. *Int Aff*, *90*(3), 625–646.

Chauhan, G. (2018). *AI safety*. Towards Data Science. https://towardsdatascience.com/ai-safety-9aeb9ca42907#:~:text=AI%20Safety%20is%20collective%20termed,of%20real%2Dworld%20AI%20systems

Cooter, R. D. (2000). Three effects of social norms on law: Expression, deterrence, and internalization. *Oregon Law Rev*, *79*, 1–22.

Deci, E. L. (1971). Effects of externally mediated rewards on intrinsic motivation. *J Pers Soc Psychol*, *18*, 105–115.

DoD. (2012, May 11). *Department of Defense standard practice: System safety* (MIL-STD 882E). Pentagon.

Eden, A. H., Moor, J. H., Soraker, J. H., Steinhart, E. (2013). *Singularity hypotheses: A scientific and philosophical assessment*. Springer.
Etherington, D. (2012). Elon Musk says all advanced AI development should be regulated, including at Tesla, Tech Crunch. https://techcrunch.com/2020/02/18/elon-musk-says-all-advanced-ai-development-should-be-regulated-including-at-tesla/
Everitt, T. (2018). *Towards safe artificial general intelligence* [Doctoral thesis, Australian National University]. https://www.tomeveritt.se/papers/2018-thesis.pdf
Figueroa, R., Zeng-Treitler, Q., Kandula, S. & Ngo, L. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, *12*(8).
Foody, G., McCulloch, M., & Yates, W. (1995). The effect of training set size and composition on artificial neural network classification. *International Journal of Remote Sensing*, *16*(9), 1707–1723. https://doi.org/10.1080/01431169508954507
Griffith, E. (2016). Who will build the next great car company? *Fortune*. http://fortune.com/self-driving-cars-silicon-valley-detroit
Gurney, J. K. (2013). Sue my car not me: Products liability and accidents involving autonomous vehicles. *Univ Ill J Law Technol Policy*, *2013*(2), 247–277.
Hamilton, I. (2020) Elon Musk slams Microsoft exclusively licensing OpenAI's text-generating software, Business Insider. https://www.businessinsider.co.za/elon-musk-criticizes-microsoft-exclusively-licensing-gpt-3-2020-9
Joint Software Systems Safety Engineering Workgroup. (2010). *Joint software systems safety engineering handbook* (SSSEH v1.0). Pentagon.
Joint Software Systems Safety Engineering Workgroup. (2017). *Software system safety implementing process and tasks supporting MIL-STD-882E* (JS-SSA-IF Rev. A). Pentagon.
Jones, M. (2019). Artificial intelligence (AI): The need for new safety standards and methodologies (ISSC37-21). *37th International System Safety Conference*.
Joy, B. (2000). Why the future doesn't need us. *Wired*, *8*(4), 238–263.
Kim, Y., Sidney, J., Buus, S., Sette, A., Nielsen, M., & Peters, B. (2014). Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions. *BMC Bioinformatics*, *15*(241).
Krantz, D. H., Peterson, N., Arora, P., Milch, K., & Orlove, B. (2008). Individual values and social goals in environmental decision making. In T. Kugler, J. C. Smith, T. Connolly, & Y. J. Son (Eds.), *Decision modeling and behavior in complex and uncertain environments* (pp. 165–198). Springer.
Kunda, Z. (1990). The case for motivated reasoning. *Psychol Bull*, *108*(3), 480–498.
Lapinski, M. K., & Rimal, R. N. (2005). An explication of social norms. *Commun Theory*, *15*(2), 127–147.
Lewandowsky, S., Cook, J., Oberauer, K., Brophy, S., Lloyd E. A., & Marriott, M. (2015). Recurrent fury: conspiratorial discourse in the blogosphere triggered by research on the role of conspiracist ideation in climate denial. *J Soc Political Psychol*, *3*(1), 142–178.
Lin, P., Abney, K., & Bekey, G. A. (Eds.). (2011). *Robot ethics: The ethical and social implications of robotics*. MIT Press.
McGinnis, J. O. (2010). Accelerating AI. *Northwest Univ Law Rev*, *104*, 366–381.
Moses, L. B. (2007). Recurring dilemmas: The law's race to keep up with technological change. *Univ Ill J Law Technol Policy*, *2007*(2), 239–285.
National Institute of Standards and Technology. (2019). *U.S. leadership in AI: A plan for federal engagement in developing technical standards and related tools*. U.S. Department of Commerce. https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf
Ozlati, S., & Yampolskiy, R. (2017). *The formalization of AI risk management and safety standards* (WS-17-02). AAAI Workshop on AI, Ethics, and Society. https://www.aaai.org/ocs/index.php/WS/AAAIW17/paper/viewFile/15175/14655
Pei, K., Cao, Y., Yang, J., & Jana, S. (2017). DeepXplore: Automated whitebox testing of deep learning systems. *Proceedings of the 26th Symposium on Operating Systems Principles*, 1–18. https://doi.org/10.1145/3132747.3132785
Porter, D., McAnally, M., Bieber, C., Wojton, H., & Medlin, R. (2020). *Trustworthy autonomy: A roadmap to assurance: Part 1. System effectiveness* (IDA Document P-10768-NS). Institute for Defense Analyses. https://www.ida.org/-/media/feature/publications/t/tr/trustworthy-autonomy-a-roadmap-to-assurance/p-10768.ashx

Radio Technical Commission for Aeronautics. (2012). *Software considerations in airborne systems and equipment certification* (DO-178C). Federal Aviation Administration.
Rodríguez-Pérez, R., Vogt, M., & Bajorath, J. (2017). Influence of varying training set composition and size on support vector machine-based prediction of active compounds. *Journal of Chemical Information and Modeling*, *57*(4), 710–716. https://doi.org/10.1021/acs.jcim.7b00088
Russell, S., Dewey, D., & Tegmark, M. (2015). Research priorities for robust and beneficial artificial intelligence. *AI Mag*, *36*(4),105–114.
Schmelzer, R. (2019). What happens when self-driving cars kill people? *Forbes*. https://www.forbes.com/sites/cognitiveworld/2019/09/26/what-happens-with-self-driving-cars-kill-people/?sh=7632fa27405c
Shneiderman, B. (2016). Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(48), 13538–13540.
Shneiderman, B. (2020). Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered AI systems. *ACM Transactions on Interactive Intelligent Systems*, *10*(4). https://doi.org/10.1145/3419764
Toloşi, L., & Lengauer, T. (2011). Classification with correlated features: Unreliability of feature ranking and solutions. *Bioinformatics*, *27*(14), 1986–1994. https://doi.org/10.1093/bioinformatics/btr300
Wiggers, K. (2019). *11 companies propose guiding principles for self-driving vehicles*. VentureBeat. https://venturebeat.com/2019/07/02/self-driving-car-report-safety-first-for-automated-driving/
Wiggers, K. (2020) Waymo's driverless cars were involved in 18 accidents over 20 months VentureBeat https://venturebeat.com/2020/10/30/waymos-driverless-cars-were-involved-in-18-accidents-over-20-month/
Wiggers, K. (2019) 11 companies propose guiding principles for self-driving vehicles VentureBeat https://venturebeat.com/2019/07/02/self-driving-car-report-safety-first-for-automated-driving/
Moses, LB (2007) Recurring dilemmas: the law's race to keep up with technological change. Univ Ill J Law Technol Policy 2007(2):239–285
Cooter, RD (2000) Three effects of social norms on law: expression, deterrence, and internalization. Oregon Law Rev 79:1–22
McGinnis, JO (2010) Accelerating AI. Northwest Univ Law Rev 104:366–381
Russell, S, Dewey, D, Tegmark, M (2015) Research priorities for robust and beneficial artificial intelligence. AI Mag 36(4):105–114
Baum, S. (2017) On the promotion of safe and socially beneficial artificial intelligence. AI & Soc 32, 543–551 (2017). https://doi.org/10.1007/s00146-016-0677-0
Kunda, Z (1990) The case for motivated reasoning. Psychol Bull 108(3):480–498
Yampolskiy, R. (2020) On Controllability of AI. arXiv:2008.04071v1. https://arxiv.org/ftp/arxiv/papers/2008/2008.04071.pdf
Chauhan, G. (2018) AI Safety. .
Deci, EL (1971) Effects of externally mediated rewards on intrinsic motivation. J Pers Soc Psychol 18:105–115
Krantz, DH, Peterson, N, Arora, P, Milch, K, Orlove, B (2008) Individual values and social goals in environmental decision making. In: Smith JC, Connolly T, Son YJ (eds) Kugler T. Decision modeling and behavior in complex and uncertain environments New York, Springer, pp 165–198
Armstrong, S., Bostrom, N. & Shulman, C. Racing to the precipice: a model of artificial intelligence development. AI & Soc 31, 201–206 (2016). https://doi.org/10.1007/s00146-015-0590-y
Borrie, J (2014) Humanitarian reframing of nuclear weapons and the logic of a ban. Int Aff 90(3):625–646
Bohannon, J (2015) Fears of an AI pioneer. Science 349(6245):252\ Joy B (2000) Why the future doesn't need us. Wired 8(4):238–263
Lin, P, Abney, K, and Bekey, GA (eds) (2011) Robot ethics: the ethical and social implications of robotics. MIT Press, Cambridge
Ozlati, S., & Yampolskiy, R. (2017). The Formalization of AI Risk Management and Safety Standards. AAAI Workshop, AI, Ethics, and Society WS-17-02.
National Institute of Standards and Technology. (2019). U.S. LEADERSHIP IN AI: A Plan for Federal Engagement in Developing Technical Standards and Related Tools (Response to Executive Order 13859). Washington D.C.: US Department of Commerce.

Defense Standardization Program Office. (2012) System Safety (MIL-STD 882E). Washington, D.C.: Pentagon.
Radio Technical Commission for Aeronautics. (2012). Software Considerations in Airborne Systems and Equipment Certification. (DO-178C). Washington D.C.: Federal Aviation Administration.
Joint Software Systems Safety Engineering Workgroup. (2010) Joint Software Systems Safety Engineering Handbook (SSSEH v1.0). Washington, D.C.: Pentagon.
Joint Software Systems Safety Engineering Workgroup. (2017) Software System Safety Implementing Process and Tasks Supporting MIL-STD-882E (JS-SSA-IF Rev. A). Washington, D.C.: Pentagon.
Lapinski, MK, Rimal, RN (2005) An explication of social norms. Commun Theory 15(2):127–147
Everitt, T. (2018) Towards Safe Artificial General Intelligence. PhD thesis, Australian National University, pages 20-21.
Lewandowsky, S, Cook, J, Oberauer, K, Brophy, S, Lloyd EA, Marriott M (2015) Recurrent fury: conspiratorial discourse in the blogosphere triggered by research on the role of conspiracist ideation in climate denial. J Soc Political Psychol 3(1):142–178
Shneiderman, B., (2020). Bridging the Gap between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-Centered AI Systems, ACM Transactions on Interactive Intelligent Systems 10, 4
Shneiderman, B., (2016) Opinion: The dangers of faulty, biased, or malicious algorithms requires independent oversight, Proceedings of the National Academy of Sciences of the United States of America. https://doi.org/10.1073/pnas.1618211113
Porter, D., McAnally M., Bieber, C., Wojton, H., Medlin, R. (2020) Trustworkthy Autonomy: A Roadmap to Assurance Part 1: System Effectiveness. IDA Dcoument P-10768-NS, Log: H 2019-000369.
Jones, M. (2019) Artificial Intelligence (AI) - the Need for New Safety Standards and Methodologies. 37th International System Safety Conference, ISSC37-21.
Eden AH, Moor JH, Soraker JH, Steinhart E (2013) Singularity hypotheses: a scientific and philosophical assessment. Springer, Berlin
Amodei, D., Olah, C., Steinhardt, J., Christiano, P.F., Schulman, J., & Mané, D. (2016). Concrete Problems in AI Safety. ArXiv, abs/1606.06565.
Foody, G., McCulloch, M. & Yates, W. (1995) The effect of training set size and composition on artificial neural network classification, International Journal of Remote Sensing, 16:9, 1707-1723, DOI: 10.1080/01431169508954507.
Pei K., Cao Y., Yang J., Jana S. (2017) DeepXplore: Automated Whitebox Testing of Deep Learning Systems, ACM ISBN 978-1-4503-5085-3/17/10.
Rodríguez-Pérez R., Vogt M., and Bajorath J. (2017) Influence of Varying Training Set Composition and Size on Support Vector Machine-Based Prediction of Active Compounds, Journal of Chemical Information and Modeling 2017 57 (4), 710-716 DOI: 10.1021/acs.jcim.7b00088
Kim, Y., Sidney, J, Buus, S., Sette1 A., Nielsen M., and Peters B. (2014) Dataset size and composition impact the reliability of performance benchmarks for peptide-MHC binding predictions, BMC Bioinformatics 2014, 15:241.
Baum, SD, Goertzel, B, Goertzel, TG (2011) How long until human-level AI? Results from an expert assessment. Technol Forecast Soc Change 78(1):185–195
Figueroa, R., Zeng-Treitler, Q., Kandula, S. and Ngo, L. (2012) Predicting sample size required for classification performance. BMC Medical Informatics and Decision Making 12: 8
Toloşi, L., Lengauer, T. (2011) Classification with correlated features: unreliability of feature ranking and solutions, Bioinformatics, Volume 27, Issue 14, Pages 1986–1994, https://doi.org/10.1093/bioinformatics/btr300

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net