22 July 24

MEMORANDUM FOR RECORD

To: Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E))
From: DoD Joint Weapon Safety Working Group (JWSWG)

Subj: Endorsement of White Paper Guidance on Level of rigor (LOR) Objectives for Supervised Learning Models in Safety Significant Applications, 03 January 2023

Ref: (a) DoD Instruction 5000.69 "DoD Joint Services Weapon and Laser System Safety Review Processes" of 9 Nov 2011
(b) Military Standard 882E "Department of Defense Standard Practice for System Safety", 11 May 2012

Encl: (1) White Paper Guidance for Preferred Level Of Rigor Activities For Weapon Systems With Supervised Machine Learning Capabilities, 15 February 2024

Reference (a) establishes policy and assigns responsibilities for the Department of Defense (DoD) Joint Services Weapon and Laser System Safety Review Processes. Reference (a) requires Joint Service safety reviews for weapon and laser systems that will be used by two or more DoD Components and establishes a DoD Joint Weapon Safety Working Group (JWSWG) that will coordinate and liaise with the DoD Laser System Safety Working Group (LSSWG) on joint safety review processes. Additionally, reference (a) authorizes publication of supporting guidance to provide specific information on the DoD Joint Services weapon and laser system safety processes.

In June 2022, the DoD Joint Weapon Safety Working Group (JWSWG) endorsed the "White Paper Guidance to Perform Functional Hazard Analysis for Weapon Systems with Artificial Intelligence Capabilities, 19 May 2022". This Level of rigor (LOR) Objectives for Supervised Learning Models in Safety Significant Applications is a continuation of that work identifying the LOR objectives that need to be performed once a Safety Function Criticality Index (SFCI) has been identified. The set of LOR objectives provided here is only applicable to Supervised Learning. Reinforcement Learning is sufficiently different in how it is developed that a separate LOR set of objectives will be required. This is future work for the Joint ML Working Group.

Subj: Endorsement of White Paper Guidance on Level of rigor (LOR) Objectives for Supervised Learning Models in Safety Significant Applications, 03 January 2023

The Level of rigor (LOR) Objectives for Supervised Learning guidance provided in enclosure (1) is endorsed by the undersigned as Co-Chairs of the DoD JWSWG. Based on this endorsement, the DoD JWSWG Co-Chairs recommend formal processing and issuance of enclosure (1).

MCCREARY.SHAWNA.M.1230236190
Digitally signed by MCCREARY.SHAWNA.M.1230236190
Date: 2024.08.01 13:23:27 -04'00'

Ms. Shawna M. McCreary (SES)
(Navy, WSESRB Chair),
Co-Chair, DoD Joint Weapon Safety Working Group

Digitally signed by HAMILTON.IAN.T.1257726874
Date: 2024.07.22 09:33:14 -04'00'

Mr. Ian T. Hamilton (SSTM)
(Army, AWSRB Chair),
Co-Chair, DoD Joint Weapon Safety Working Group

Digitally signed by LAZAR.ANDREW.T.1235651544
Date: 2024.07.31 11:13:31 -06'00'

Col Andrew T. Lazar
(Air Force, NNMSRB Chair),
Co-Chair, DoD Joint Weapon Safety Working Group

White Paper

# PREFERRED LEVEL of RIGOR ACTIVITIES TABLE
For
# WEAPON SYSTEMS WITH SUPERVISED MACHINE LEARNING CAPABILITIES

15 February 2024



POC: Jerome P (Jay) Ball, jerome.p.ball.civ@us.navy.mil

Enclosure (1)

## Accreditations

Jerome (Jay) P Ball
Naval Ordnance Safety and Security Activity (NOSSA)
Deputy Executive Director/CHENG

Gunendran Sivapragasam
Technology System Safety Technical Lead
Technology Integration Safety Branch (R44) NSWCDD Dam Neck Activity

Anh Belanger
System Safety/Software System Safety Lead
U.S. Army Aviation and Missile Command Redstone Arsenal, AL

Benjamin Schumeg
Emerging Technology SW Q/R/S
QE&SA / DEVCOM Armaments Center (FCDD-ACE-QWF)
U.S. Army Futures Command

Tarek Abulmagd
System Safety Engineering (FCDD-ACE-QSC)
U.S. Army Futures Command

Benjamin Werner
Technical Lead, Quality Engineering and Systems Assurance Directorate
DEVCOM Armaments Center
U.S. Army Futures Command

Special thanks to Mr. Bruce Nagy, NAWCWD China Lake, whose contributing works played a significant role in the development of the LOR objectives.

# I. LEVEL OF RIGOR (LOR) OBJECTIVES FOR SUPERVISED LEARNING MODELS IN SAFETY SIGNIFICANT APPLICATIONS

In June 2022, the DoD Joint Weapon Safety Working Group (JWSWG) endorsed the "White Paper Guidance to Perform Functional Hazard Analysis for Weapon Systems with Artificial Intelligence Capabilities, 19 May 2022". This white paper is a continuation of that work, identifying the LOR objectives that need to be performed once a Safety Function Criticality Index (SFCI) has been identified for Machine Learning (ML) functions. This white paper specifically addresses Supervised Learning.

Figure 1 represents the complimentary approaches that address both the ML lifecycle development (ML Assurance) and ML hazard analysis process (System and ML Safety). Like software safety, it is necessary that these two approaches work together throughout the lifecycle development to produce ML with reasonable confidence for the application. ML LOR objectives should consist of both assurance tasks that guide the development of the ML capability and hazard analysis tasks that identify and mitigate hazards found throughout the lifecycle of the capability.

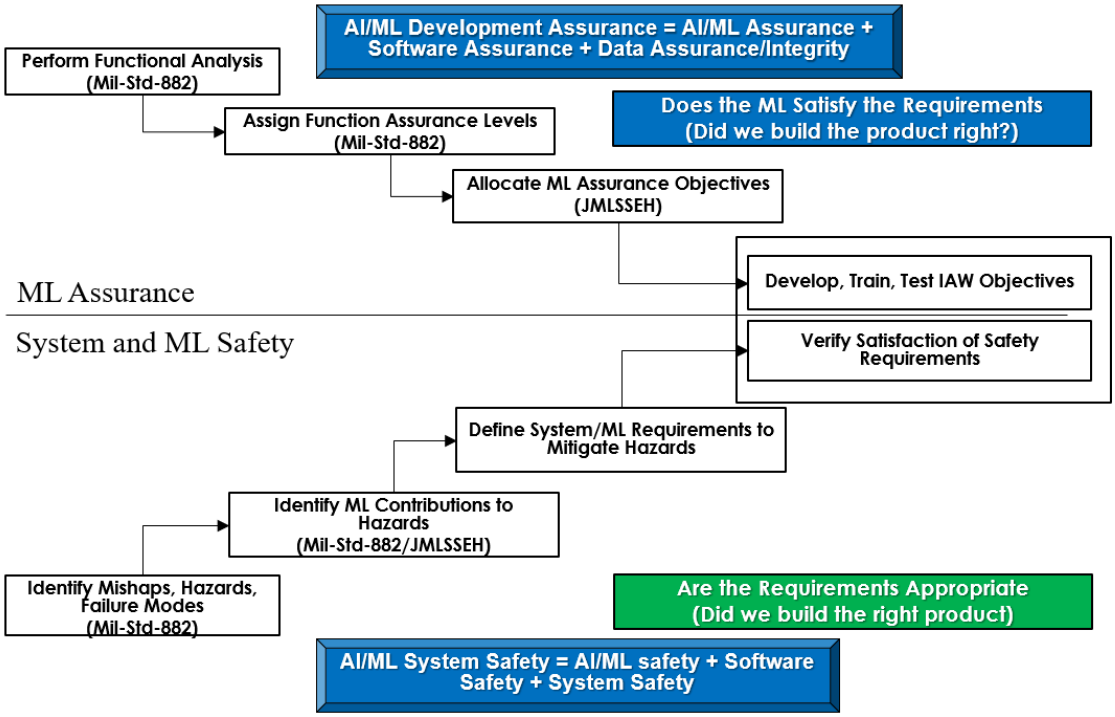


Figure 1. ML System Safety and Assurance Relationship

The focus of this paper is to detail the LOR objectives for the Assured Model. The Assured Model being a model that has had appropriate LOR applied to provide confidence that contributions to hazardous conditions throughout development and deployment have been controlled and mitigated to an appropriate level. Each aspect of the model lifecycle development activities and processes must be accounted for in the LOR. Figures 2, 3, and 4 provide examples of various developmental frameworks and processes that have informed the identification of LOR objectives.

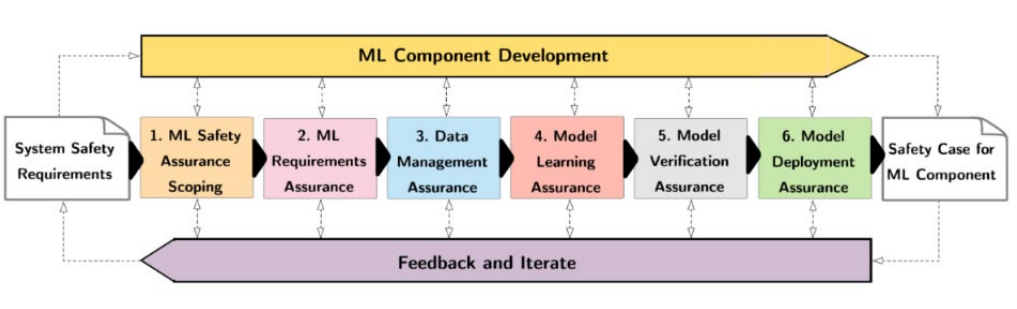


Figure 2. AMLAS ML Assurance Process (Ref 1)

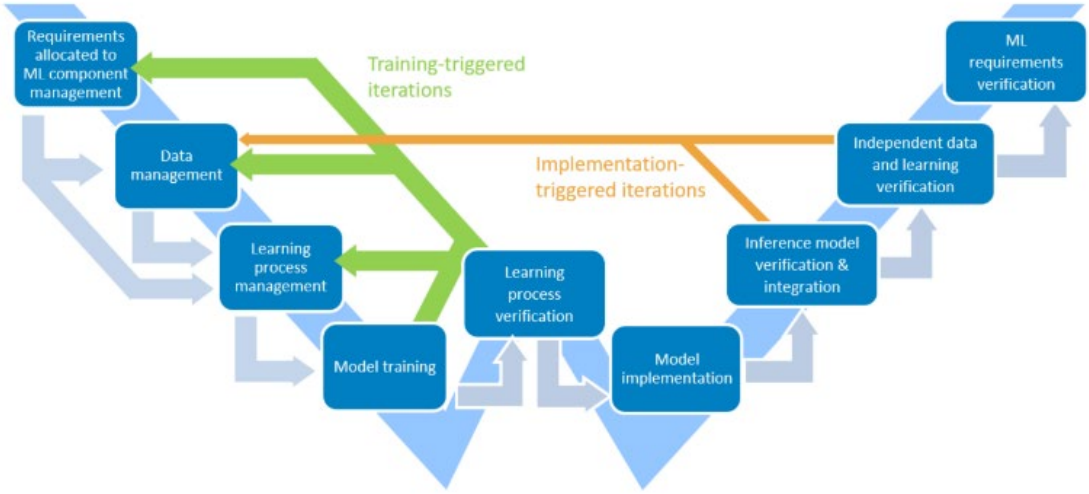


Figure 3. Learning Assurance W-shaped Process (Ref 2)

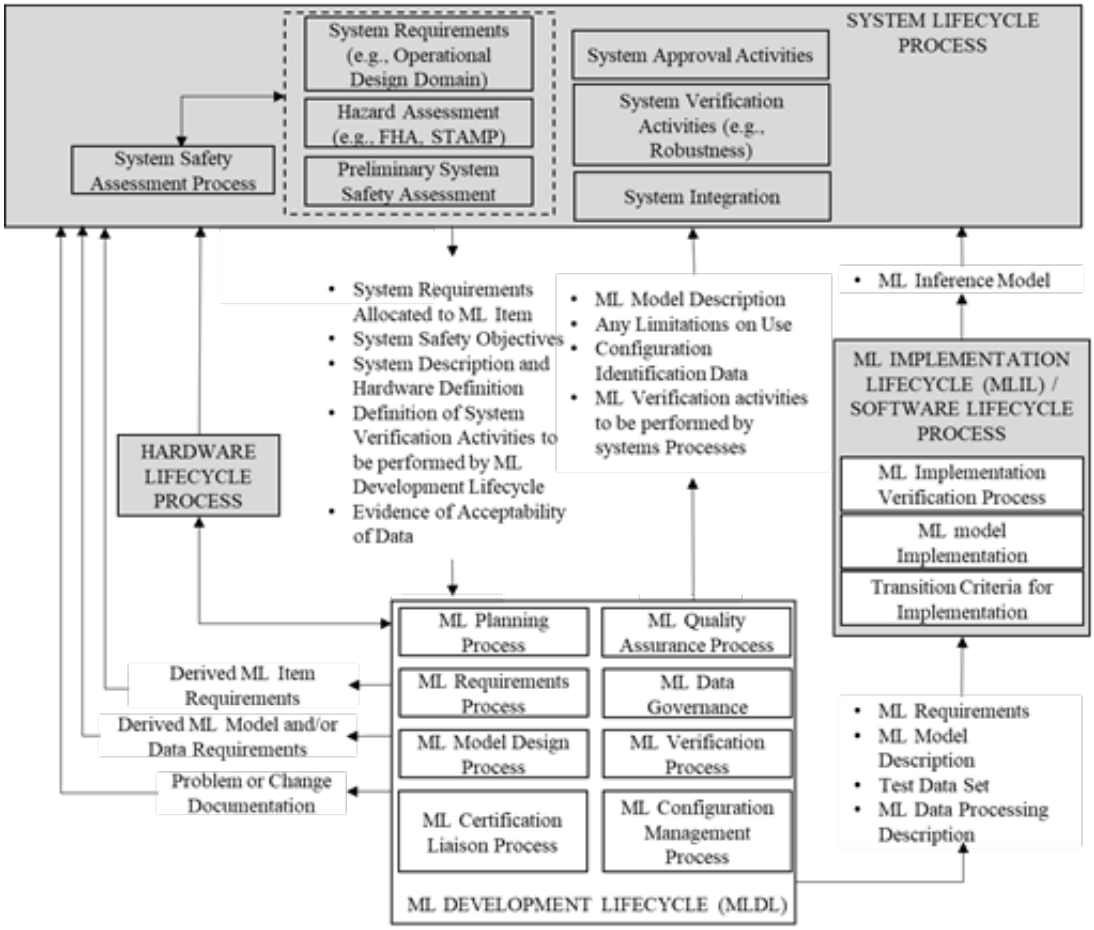


Figure 4. ML Development lifecycle (Ref 3)

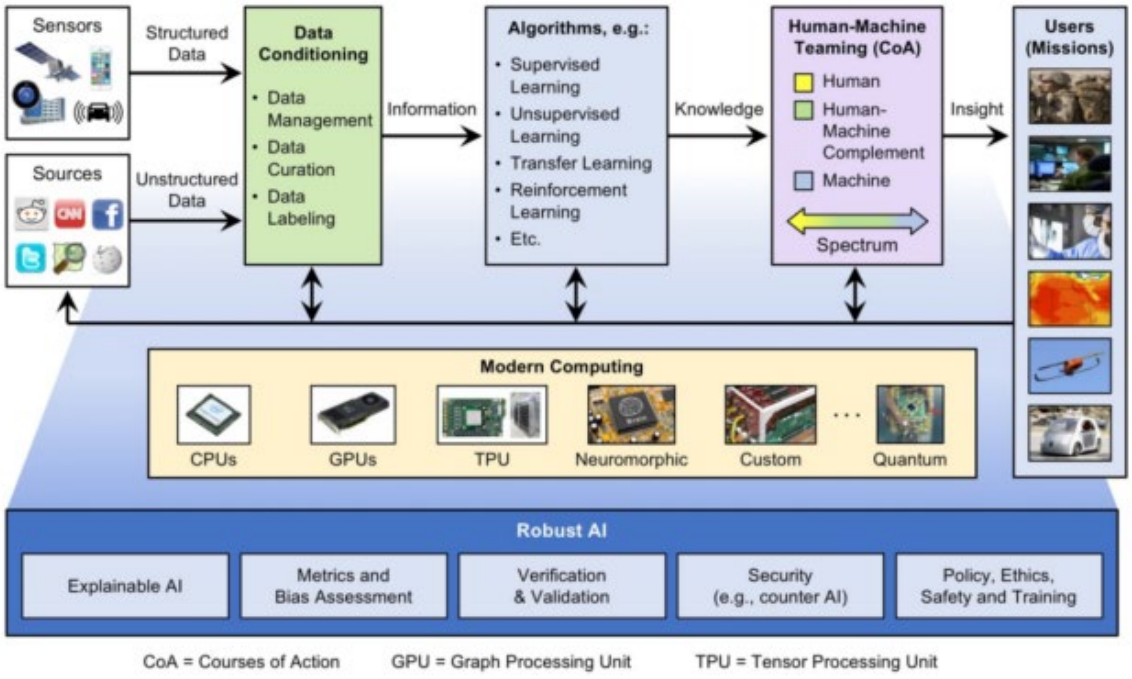


Figure 5. ML Architecture (Ref 4)

The LOR Objectives are provided as Appendix A to this paper. This set of LOR objectives is only applicable to Supervised Learning. Reinforcement Learning and other ML types are sufficiently different in how they are developed that a separate LOR set of objectives will be required. This is future work for the Joint ML Working Group.

The LOR objectives are grouped to loosely fit into the MIL-STD-882E construct. They address Requirements, Architecture, Design, Data, Algorithm, Coding, Testing and Evaluation (T&E), and Human Machine Teaming (HMT). Each task is intended to provide some specific information or to control specific types of errors that may be introduced into the lifecycle of a ML capability.

A key insight regarding the safe deployment of ML is built into the structure of the LOR matrix included in Appendix B. The uncertainty associated with ML precludes the ability to provide sufficient confidence that ML could be deployed in a function without interlocks based solely on developmental assurance, thus this LOR matrix does not contain

a column for SFCI 1 objectives. This depicts the challenges of ensuring safe ML that can be likened to a two-sided coin, managing system complexity on one hand and uncertainty on the other. The view of the joint service team is that ML used in SFCI 1 Functions cannot be accomplished with the application of LOR alone without some residual risks.

Much is lost in the ability to deeply analyze and assure behavior when going from traditional software to ML. It is not possible to create artifacts and information that enable low level design hazard analysis and code level hazard analysis. Additionally, it is not possible to conduct Requirements Based Structural Coverage Analysis at the Modified Condition/Decision Coverage (MC/DC) level as well as data and control coupling. These analyses provide a deep understanding of implemented behavior in traditional software but nothing commensurate exists with ML. Current efforts at machine learning explainability are far off from providing commensurate insight.

Lastly, the Operational Design Domain (ODD) will always be out of alignment to some degree with the Data Distribution of the training set. By the nature of the process, the training Data Distribution is a sample or subset of the ODD, and while the goal is robust generalization, this will always lead to some remaining uncertainty and an expected lower success rate in deployment in the operational environment than in testing. Certification agents or safety review boards should seek to understand how this uncertainty is reflected in risk assessments and the measure of confidence in the ML model.

This is not to indicate that ML cannot be deployed in SFCI 1 functions, just that LOR alone is insufficient to mitigate risk to a level comparable to that achieved by applying LOR to traditional software. Program Managers and risk acceptance authorities should be aware of these limitations when making decisions between capabilities and safety.

Confidence measures without the application of LOR will provide less aligned measures of a model's ability to generalize, therefore it is not sufficient to rely solely on confidence measures to support a deployment decision for ML. Like software, deployment decisions for a given SFCI and associated risk level are based on the combination of LOR, validation, and measured confidence.

Like any other function in system safety, the ability to reduce an SFCI based on an interlock requires that both functions meet the designated SCFI level. In other words the originating function and the interlocking function must both have evidence of appropriate LOR. For example, to move an originating function from SFCI 1 to SCFI 2 requires both the originating and interlocking functions to be developed to SFCI 2. If the originating function does not have LOR applied the interlocking function must be considered to inherit the criticality of the originating function without the interlock. For example, an SFCI 1 function without LOR is interlocked with a secondary function, the secondary function must be shown to meet SFCI 1 LOR requirements and cover all failure conditions of the originating function.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX A - LEVEL OF RIGOR OBJECTIVES

Acquisition programs should consider ML as a proposed solution for implementation of one or more safety- significant functions of a system only when there is a clear reason to avoid a traditional solution. ML algorithms are not transparent or verifiable by traditional software safety means. Using ML for safety-significant functions introduces new risks that must be mitigated by both the development process and system design. The following list of objectives aid to mitigate the lack of transparency, domain shifts, alignment problems, and unintended correlations mistaken as causation that are especially problematic without explainable or interpretable ML algorithms.

## A. REQUIREMENT ANALYSIS TASKS

System engineers should be aware that the use of ML has a myriad of attendant assurance issues; the following Requirements Analysis Tasks have been developed to document rationale for the use of ML in safety significant functions and influence the development process of the ML under consideration.

**RA - 1.** Assess and document justification for the proposed use of ML over operator input, more traditional software, firmware, or hardware techniques. Provide rationale as to why an ML solution is a "better" fit to the functional requirement.

ML inherently will have some uncertainty in its outputs. Some ML models are effectively black boxes. Assess and document rationale for the proposed use of ML model in the design of a safety significant function over more traditional software, firmware, or hardware techniques in terms performance improvements, added uncertainty, and unexplainability.

Questions to address when choosing ML as a solution include, but are not limited to:

1 - Can this function be performed with a non- ML solution?

2 - What are the potential failure modes associated with the ML solution versus the non- ML solution and associated risks?

3 - What are the relative benefits of both the ML solution and the non-ML solution.

4 - Is one solution more complex than the other? Is that added complexity justified by the perceived benefits?

**RA - 2.** Document justification (selection criteria) of selected supervised ML technology or algorithm (Naive Bayes, random forest, etc.) that is the most appropriate for use in this function e.g., would this algorithm provide for the best operational performance.

Document the selection criteria for the optimal ML algorithm for the given implementation. Each algorithm comes with its own inherent issues and benefits, consider these when determining the selection criteria. Topics to address include:

1 - Is the algorithm that was identified the best among considered options for this function within the system? Provide rationale.
2 - Was the choice of algorithm based on what provides the best operational performance and understanding of operational limits? How was that determined?
3 - How are the operational limits of this machine learned function identified?
4 - Include in the assessment discussion of possible hazards based on the trade-offs between algorithms considered.

**RA - 3.** Document how performance for this ML model is measured e.g., success rate (% confidence or similar).

Determine and document how performance for this ML model is measured e.g., success rate for a particular operational environment. Metrics to measure performance (success rate) of an ML model vary depending on the function it performs. Some examples are the Confusion Matrix as an evaluation metric for a classification function and Mean Absolute Error for a regression function. The success rate or the measure of success should be accounted for in hazard analysis and risk assessments.

Some considerations that should be addressed in the documentation:

1 - What is the intended output?
2 - What does ‘success’ mean for the implementation?

3 - What is success being measured against?

4 - What is the ODD?

5 - Are we getting intended results that are based on unintended learning/characteristics?

**RA - 4.** Document the Change Management (CM) process for the Datasets (Training, Test and Validation), testing method, hyper-parameters, data integrity, model, and the algorithm(s).

From a system safety perspective, it is very important that the behavior of a trained algorithm is repeatable and consistent. The process and activities to manage the different datasets and their data under CM should be defined, including the aspects related to the tracking and management of changes, and the addition of data after the certification or approval of the ML system. A proper CM will allow test results to be repeatable, allowing a better understanding of trained algorithm behavior. To enable this, the CM process must encompass the following items:

1 - Datasets. For example, the incoming in-service operational data should be traceable to their origin.

2 - Order of the individual instances within the training set (order of input for training data).

3 - Hyper-parameters.

4 - Characteristics of the Data, e.g., features, pixel count etc.

5 - Format of data.

6 - Random number generator seed variables.

7 - How data is protected from corruption or unintended modification e.g., verification to ensure data has not been inadvertently altered.

8 - Other relevant information to ensure the results of the algorithm are consistent when retrained with the same parameters.

**RA - 5.** Ensure ML development environment and tools are addressed in the Software Development Plan and the appropriate LOR is applied.

Software tools used in development of the ML function are safety significant if they are used to implement a safety significant function. Conduct software LOR on these tools according to the SFCI assigned to the function. Consider approaches such as DO-330 or ISO-IEC 62304.

Additionally, ML tools are sometimes used to develop synthetic training or test data. How these tools are developed, qualified, verified, validated, and accredited needs to be clarified and documented as errors in these tools will significantly affect the behavior of the ML function.

**RA - 6.** Ensure that the System Safety Program Plan includes sections for ML System Safety.

Much like software system safety, ML system safety should have an independent section that discusses the interrelationships between the System Safety program, the Software System Safety Program, hazard analysis to be accomplished, LOR requirements, and System Safety Working Group (SSWG) activities. Identify key members and stakeholders for ML SSWGs such as Fleet Representatives, ML Developers, T&E, Software Safety, Software Engineering, and Systems Engineering.

## B. ARCHITECTURE ANALYSIS TASKS

ML developers must consider the impact of an ML subsystem on overall system safety and overall system impacts to the ML subsystem safety. ML functionality has unique vulnerabilities e.g., the need for accurate data, which must be accounted for when evaluating the ML function within the overall architecture of the system.

**AA - 1.** Document ML Datasets source architecture and data architecture e.g., modality type.

When creating Datasets, it is important to understand the operational environment being represented to ensure adequate training of the ML algorithms. The system

architecture will prescribe the modality of data necessary to train the model. Training data is either found from live events, or synthetically created to match the operational scenario. The system architecture details what sensors, human input and other sources of data are provided to the model and includes the attributes/features from each of those sources. Understanding the relationship between the system source data architecture and the ML training data architecture is fundamental to developing a reliable ML system (Ref. 5). It is vital that the training data includes all modalities of data that will be provided in the operational environment.

Modality refers to the data source type, for instance source data such as text, radar pings, image or video feed, or GPS would represent different modalities of operational source data provided to the model.

Data source architecture refers to the design of the systems that store data and provide data for the model to access.

**AA - 2.** Justify the SFCI of ML function. Determine if other control entities (such as a human operator, software, firmware, hardware) can be inserted to reduce the Function Control Category (FCC).

Identify hardware, software, or human interlocks that can prevent a mishap if this function were to fail. As much as possible, investigate interlocks that could be used to reduce the FCC (autonomy of the ML function). Having a higher criticality FCC assignment (fewer controls/interlocks) will result in additional LOR applied to the function, increasing development time and cost.

## C. DESIGN ANALYSIS TASKS

Due to the unique limitations of ML i.e., the inherent uncertainty contained within the outputs of the algorithm, it is important to ensure the algorithms are being provided with Datasets that properly represents the operational environment to reduce output errors when the resultant model is placed in the operational environment. Developers need to understand the limitations of the algorithm, and how the model will behave when faced with low quality or unexpected inputs.

**DA - 1.** Determine if the system design includes quality check mechanisms to ensure the integrity of the input data, ensuring that corrupted or incorrectly formatted data is not provided to the deployed Model.

Document how the quality check mechanisms (e.g., checksum, CRC) will ensure the input data is robust. Document the error rate metrics. Metrics that are more robust are necessary for functions that are more critical, e.g., CRC for SFCI 1&2, Checksum for SFCI 3 and lower.

**DA - 2.** Document behavior of the ML function when the algorithm experiences an operational environment that is beyond the limits of the Datasets. What does the function do when faced with input it is not trained for?

Determine responses of the ML when presented with data outside the ODD to identify failure modes. Document what the model outputs when the system is placed outside its operational environment. The focus is on operational drift scenarios where the model is being presented with data from the environment it was not trained to handle.

**DA - 3.** Determine if there are 'wrappers' around the function that are verifying and/or intervening when outputs of the ML function are unreasonable e.g., outside a defined range, unintelligible.

Document how the non-ML method e.g., human or a runtime assurance application are used to “wrap” and/or “watchdog” the ML component. The concept is to ensure that outputs from the model do not exceed defined safety parameters. For example, what happens if the ML model confidence rate of identifying a track falls below a pre-defined threshold (e.g., 50%), control commands exceeding the design parameters of the system (excessive roll rate or climb rate) etc.

**DA - 4.** Document how the Application Programming Interfaces (APIs)/ Structured Query Language (SQL)/Message (MSG) use and interface is verified.

Document the data assurance and integrity process. Errors in the API/SQL/MSG could result in significant issues e.g., corrupted, or erroneous data provided to training or operational databases from sensor sources. Any time tools are developed or used off the shelf to manipulate, store, or transfer data there is the potential to corrupt the data.

APIs are commonly used in ML for data retrieval, data processing, and model deployment. Below are some examples of how APIs are used in machine learning:

1 - Data Retrieval: APIs can be used to access data from external sources. ML models require large amounts of data for training and APIs can facilitate the process of acquiring data.

2 - Data Processing: APIs can also be used for data processing, which involves data cleaning, filtering, and transforming. For example, image processing APIs can be used to resize or crop images.

3 - Model Deployment: APIs are also commonly used for model deployment. This allows end-users to access ML models through a simple interface, without needing to understand the underlying complexity of the model.

SQL: is a domain-specific programming language used to manage and manipulate relational databases. SQL is commonly used in ML for data preprocessing and data storage. Some examples of how SQL is used in ML:

1 - Data Preprocessing: ML models require large amounts of data for training. Before data can be used for training, it needs to be preprocessed, which involves data cleaning, filtering, and transforming. SQL can be used to preprocess data stored in relational databases. SQL queries can be used to remove duplicates, filter data based on specific criteria, and join tables.

2 - Data Storage: ML models require access to large amounts of data, which can be stored in databases. Relational databases are often used to store structured data, such as customer information, financial data, and inventory data. SQL can be used to retrieve data from databases for use in ML models.

3 - Model Evaluation: After a ML model has been trained, it needs to be evaluated on a test dataset to assess its performance. SQL can be used to extract data from databases for model evaluation. For example, SQL queries can be used to select a

subset of data for testing and to compute evaluation metrics, such as accuracy, precision, and recall.

**DA - 5.** Document how databases housing all data (training, test, validation, and operational databases) are protected from unintentional data corruption e.g., database physical or electrical, or processing damage.

Data Integrity of data is a primary concern, it is important to document and address vulnerabilities related to the use of databases. Any databases used in the implementation or in the development environment of a ML model used within safety significant functions must be considered as safety significant.

**DA - 6.** Document that wherever possible, the design makes use of predefined and well understood modules.

The use of well documented and well understood modules will assist greatly in the characterization of how ML system will function both in predicted and unexpected operational situations.

In ML, modules refer to the pre-built libraries or packages of code that provide a set of functions and tools to perform specific tasks in the ML pipeline. These are valuable in routine problems that have a pedigree of use. Care must be taken in safety critical applications that no unnecessary functions or code are included in predefined modules.

**DA - 7.** Document that wherever possible, the design makes use of predefined and well understood architectures.

The use of well documented and well understood architectures will assist greatly in the characterization of how the ML system will function both in predicted and unexpected operational situations. The architecture of an ML model refers to its overall structure, including the number and types of layers, the activation functions used, and the connections between neurons.

The architecture of a ML model depends on the type of problem being solved and the type of data being used. Different architectures are designed to handle different types

of data, such as images, text, and sequential data. The architecture of an ML model determines how it processes the input data, extracts features, and produces an output.

**DA - 8.** Design for interoperability with human operators, human co-warfighters, and other systems.

ML systems must be designed with its operational use properly understood. If the ML system will be operating with humans or other systems, then there should be requirements adequately identified for this interoperability.

## D. DATA CURATION TASKS

Data curation is the organization and integration of data collected from various sources used to train the ML algorithm. Data curation also involves annotation, publication, and presentation of the data such that the integrity of the data is maintained over time, and the data remains available for reuse and preservation. Data curation normally supports a targeted ML goal, where the organization of the data is based on the classification or regression needs of the algorithm. The process of data curation involves the selection, creation, organization, and maintenance of data sets so that they can be accessed and utilized as needed correctly and accurately.

**DC - 1.** Identify available sources of data and determine what type of data (real or synthetic) will be used to train, test, and validate the ML function in accordance with the ODD.

Source of data availability is an important aspect to consider when collecting data to train a model. Quality and quantity of data will significantly affect the accuracy of the trained algorithm. When choosing a data source for training, questions to consider include:

1 - Have analyses been developed to document the number of configurations available and the quantity of data for each configuration within these data sources?

2 - Is the training data organized in terms of attributes/features, thereby being able to identify potential missing and sparse data occurrences from sources?

3 - If intelligence sources were used, how accurate and reliable are those sources? (Ref. 5)

Data sources may be real or synthetic, they may have been already created, or may be created/collected/developed for a particular project from a real or synthetic source. Real data is often more expensive and difficult to collect but would represent a more accurate picture of the real world. Synthetic data is often regarded as the cost effective and faster option but not the best when it comes to accuracy. Thus, care must be taken when relying on synthetic data since it does not originate from the operational space. It is recommended that analysis be conducted to document how well the synthetic data developed for training is representative of the real world and traces back to CONOPS/system requirements.

**DC - 2.** Ensure that training, testing, and validation data are appropriately identified and separated from each other.

Separating data into training, test, and validation sets is crucial for ML because it helps to prevent overfitting and ensures that the model is generalizable and robust. Here is a brief explanation of each set:

1 - Training set: This is the data that is used to train the model. The model learns from the patterns and relationships in the training data.

2 - Validation set: This is a subset of the data that is used to tune the model's hyperparameters (e.g., learning rate, regularization strength) and to evaluate the model's performance during training. The validation set helps to prevent overfitting by allowing the model to be evaluated on data that it hasn't seen before.

3 - Test set: This is a subset of the data that is used to evaluate the final performance of the model after it has been trained and tuned. The test set helps to provide an unbiased estimate of the model's performance on new, unseen data.

If data is not separated into these sets and the same data is used for both training and evaluation, the model may perform well on the training data but poorly on new, unseen data. This is because the model has learned to fit the training data too well and has not generalized well to handle new data. By using separate training, validation, and test sets, overfitting is prevented and the model's performance on new data can be more accurately estimated.

**DC - 3.** Document the type of data (real or synthetic) and sources/conditions/metadata associated with the data, identify any shortcomings.

Once the type and sources of data are determined, information about the data must be documented. This should include information about the source of the data, the conditions of the data (e.g., collection location, time, data characteristics, accuracy, diversity, sensor attributes), metadata and any shortcomings. A Data Card that includes information about origin, collection and preparation, description, label information, limitations, access, and restrictions of the data may be used.

**DC - 4.** Document how the data is prepared for the model, include how the data retains the desired characteristics for future training iterations.

Data must be prepared to ensure the model is trained, validated, and tested to develop the desired characteristics when used in the operational deployment. Data preparation may include data cleaning (removal of bad data, dealing with missing data, filtering, range and rate checks, consistency checks), feature computation, feature extraction, feature selection, dimensionality reduction, feature engineering, normalization, labeling, and bias management. An evaluation is required to ensure the desired characteristics are maintained for accurate model generation when using the data for retraining or updating of the model.

**DC - 5.** If generating synthetic data, document the data generation process.

Synthetic data refers to artificially generated data that imitates the statistical properties of real-world data. Synthetic data can be used in situations where real-world data is sensitive or confidential. Synthetic data can be used to increase the size of a dataset, which is particularly useful when working with small datasets. By generating synthetic data that expands the range of variation in the dataset, the model can be trained to be more robust and generalize better to new data. Synthetic data can be used to test the robustness of a ML model under different conditions. By generating synthetic data that simulates various scenarios, the model can be tested for its ability to handle new, unseen data. Synthetic data can be used to balance the class distribution in imbalanced datasets. By generating synthetic data for the minority class, the model can be trained on a more balanced dataset, leading to better performance in the minority class.

Synthetic data is only as good as the quality of the algorithm used to generate it. The synthetic data should be representative of the real-world data and should not introduce any biases or artifacts that could impact the performance of the model. Therefore, the generation of synthetic data requires careful consideration and validation to ensure that it provides a valid representation of the real-world data.

Questions to address:

1 - Does the synthesized data include sufficient noise for generalization in the operational environment?

2 - Is the synthetic generation applicable to operational environment inputs?

Document analysis showing that the synthetic data truly represents the operational environment. Document how bias, test variance, overfitting/underfitting have been considered.

**DC - 6.** Analyze and assess if Datasets (real or synthetic) reflect the ODD.

Datasets could be real data collected (from available sources or from a test or excursion explicitly executed to collect data) or synthetic data that is generated or synthetic data that were collected to develop the model. If synthetic data was developed, how accurate were the approximations used in creating the model? Document how closely the approximations used fit the real-world i.e., ODD.

It is important to ensure that appropriate attributes and features were included. If the algorithm is trained on simulation results, then the concern is that there is a "garbage in, garbage out" issue—poor real world representative synthetic data will result in an inferior model. The same concern could be said for poor "real" data that is not representative of the operational environment.

It is also worth noting that considerations for noise in the training dataset should be examined. Adding noise during training could improve the robustness of the model if the noise introduced doesn't present a strong learning signal to the learning model. The ratio of noisy instances should be defined so that it can be validated to provide confidence that this training data set will perform as defined. The goal is to have good quality and comprehensive training data that will result in a robust model.

**DC - 7.** Analyze and assess the attributes of the objective to determine whether the training, test, and validation data represents adequate quantity, quality, and attributes for the desired classification.

Determine if the quality and quantity of Datasets are sufficient and appropriately addresses the required attributes or feature space. Attributes are a subset of features that describe a class such as feather to describe a bird, or nose type to describe a wolf, etc. Quality refers to the correct number of attributes (including their definition as primary, secondary) that are representative of the deployed operational environment, including noise factors. Quantity refers to the number of data/instances used for training, with consideration to mix ratios, underfitting, overfitting and majority/minority classes.

Does each Class have an appropriate number of attributes, or values, which can be learned by the algorithm for the class/number being determined? In other words, has overfitting and underfitting been considered for each Class/number regarding the quantity of attributes/values simulated/collected and does that quantity reflect real world operations?

Document how the training data, attributes or values within the instances used for categorization/regression are sufficient to maximize success for data inputs not in the original training set.

**DC - 8.** By ML Class, define the rating of importance of attributes (precedence) that the ML algorithm needs in the training data to perform at the highest success rate based on the operational environment during deployment.

For each ML class, define requirements that rank the importance of attributes, i.e., creating a priority list, within each instance that the ML algorithm will be trained to recognize. This ranking represents a baseline to determine if a quality training set is being used that will produce the expected model.

As an example, a TSAT (Ref. 6) supports the requirements group's understanding of the operational environment in ranking all attributes that will be used by each class. The approach allows the project to assess the training data to determine if the requirement group's ranking is statistically like the statistically determined ranking of the training set. Statistical ranking determination is based on several occurrences of each attribute within the entire training set. The result of comparing the initial, required ranking to the statistically based ranking is calculated as a single numeric value. The single numeric value represents how well the requirements group's ranking matches the training set compositions.

**DC - 9.** Once the priority of attributes has been determined in DC-8 above, verify that the attributes are appropriately reflected in the Datasets (real or synthetic).

Once attributes are ranked in terms of priority, the next question is whether the ranking process has uncovered a grouping of attributes based on the importance and availability of data during a mission e.g., the presence of a group of primary, secondary, and tertiary attributes.

When some of the primary attributes are missing, secondary attributes may be used as input for the algorithm to produce a more successful categorization rate. This also means that a mix of primary and secondary attributes are needed as part of the training data. It

should be noted that primary attributes should occur more often than secondary in the training data, based on what is most important for the algorithm to learn.

Primary, secondary, tertiary, etc., will be based on how often a group of attributes are expected as input to the algorithm during deployment. If they were all considered primary, what happens when there is missing and sparse data issues during deployment? Thus, to support realism, should secondary and tertiary attributes be considered? If considered, what should be the ratio of primary to secondary and tertiary attributes? Can this be a requirement? Specifically, how will the priority and ratio of a grouping of attributes be determined and how will it be used for testing? As an aid to determine priority and ration of a group of attributes a starting point could be a process to create operational scenarios looking at nominal and extreme cases.

The ratios of the attributes in the training data should reflect the priorities identified. This expected ratio can be investigated using the Sources to Attribute Ratios for 1, 2 or 3(nth) (StAR-n) (Ref. 6) Order Matrix tool to compare what is required to what is provided as training data to assess enough training data per attributes.

Note that there are other proven statistical methods that have been used to investigate training data sufficiency, e.g., z-value analysis of Logistics Regression algorithm.

**DC - 10.** Have the appropriate features in the Datasets been selected to allow the ML function to perform correctly in the ODD?

Document the process for selecting features or attributes in the data and their appropriateness for the operational environment and intended function of the ML model. Features are used to represent the input data in a way that the ML model can learn from it. A feature is a measurable characteristic of the data that is used to capture its essential information. Here are a few examples of features used in ML:

1 - Text data: In natural language processing, features can be things like word frequencies, n-grams, or other text-based statistics that represent the content and structure of the text.

2 - Image data: In computer vision, features can be things like edge detectors, corner detectors, or other image-based features that represent the structure and content of the image.

3 - Tabular data: In structured data, features can be things like numerical values, categorical values, or other statistical measures that represent the patterns and relationships in the data.

The choice of features is crucial for the performance of the ML model, as it directly affects its ability to learn from the data. A good feature should be informative, meaning that it should capture the essential characteristics of the data, while at the same time being simple enough for the model to learn from. In many cases, feature engineering is a critical part of the ML process, where domain knowledge and creativity are required to identify the best set of features for the task at hand.

**DC - 11.** Assess any missing attributes or features in the Datasets with the operational community and the system safety working group.

Determine how missing attributes or features in the data that would be experienced in the operational environment would impact performance of the ML model and any associated risks.

**DC - 12.** Determine if the Datasets included instances of sufficient noise/clutter that allows the algorithm to function robustly when deployed.

Document how sufficient noise has been included in the data to ensure that the algorithm is trained to handle off-nominal conditions. Noise is variation or disturbance in the data, which impacts the performance of a ML model. Noise can have negative impacts but can improve the generalization of the model. Some ways in which noise can be included in the training data for ML:

1 - Random noise: Adding random noise to the input data can help to make the model more robust to variations in the data. For example, in image classification tasks, random noise can be added to the input image to simulate variations in lighting or camera angles.

2 - Adversarial noise: Adversarial noise refers to noise that is intentionally added to the data to fool the ML model. By training the model on adversarial examples, it can learn to be more robust, generalize better, and improve its performance on real-world data.

3 - Label noise: Label noise refers to errors in the labels of the training data. By including label noise in the training data, the model can learn to be more robust to errors in the data and generalize better to new, unseen data.

It is important to note that the inclusion of noise in the training data should be done carefully and with a clear understanding of the impact on the model's performance. Too much noise can make the model less accurate, while too little noise can lead to overfitting and poor generalization. Therefore, the amount and type of noise should be carefully chosen based on the specific task and the characteristics of the data.

**DC - 13.** Determine if the training, test, and validation data included sufficient operational complexity that allows the algorithm to function robustly when deployed.

One concern is the effects of operational complexity. The user or user representative must provide their version of operational complexity. Operational complexity can be described as the combination of environmental conditions, network and signal disruptions, obstructions, stimuli, battle space complexity; the level of stimuli occupied in the ODD that can affect the decision process.

Operational complexity can have a significant impact on the success of a ML project, as it can affect the quality, quantity, and diversity of the data used to train the model. This requires a deep understanding of the data and the problem domain, as well as a systematic approach to data collection, preparation, and management.

**DC - 14.** Document process to determine if the Datasets (real or synthetic) will result in underfitting or overfitting of each Class.

Determine that for each Class, there is an appropriate quantity of data with the right attributes, or values that can be learned by the algorithm for the Class being determined. In

other words, has overfitting and underfitting been considered for each Class/number regarding the quantity of attributes/values simulated/collected and does that quantity reflect real world operations? Overfitting results in the model that has very low tolerance for input data that is not close to the original training data input and can only be used in a very narrow scope of conditions. The model is trained so well in a certain set of data that it doesn't know anything else. Underfitting results in a model that is incapable of categorizing inputs to the correct class/approximation, the training data is too generalized. Both indicate a poor level of performance.

Assess if the training data, and the attributes or values within the data instances of the training data, is sufficient to maximize success for data inputs not in the original training set. NOTE: This assessment must evolve with how the algorithm will be operationally deployed.

**DC - 15.** Document traceability of training, test and validation data to operational conditions and operational requirements.

Ensure training data and validation data properly represents the operational environment and is traceable to the operational concept. Data sets should trace back to the operational environment outlined in the Capability Design Document/ Concepts of Operations/ Concept of Employment/ Operational View-1 documentation, or the ODD. The system requirements and requirements for supporting capabilities along with identified use cases will specify what conditions the system is expected to operate under. Datasets should trace back to these design documents to ensure the product is capable of expected performance in the intended conditions. Not only are the correct proportion of training data attributes needed, but also the proportion must be in alignment with the reality of the system being deployed. In other words, what the algorithm will experience if involved in a mission. Operational complexity could cause the system/model to perform not as intended, thus it is important to understand how the system will react to these conditions. How well does the ML algorithm support increased operational complexity and what is the impact of sparse and missing data issues?

**DC - 16.** Analyze the system and document if the Datasets (synthetic or real) included scenarios where there is sparse, inconsistent, interrupted, or missing data (e.g., data with missing attributes).

Creating a comprehensive Datasets requires the developer to assume the algorithm will need to perform in an imperfect world, where some of the primary sources for the algorithm may become unavailable and there may be missing, sparse or inconsistent. In this case secondary or tertiary sources might need to be employed to fill the gap during this period or the training data should include expected inputs of failed or faulty sensors and networks.

Missing data (Ref 5), for this purpose, is considered when the data exists but is outside acceptable parameters required to be used as input to the algorithm. For example, a radar sensor states a tanker is moving at 1000 knots. The sensor's filter would recognize this as erroneous data and would not report it as usable data for input to the algorithm, but instead, would leave the field blank. Therefore, for whatever reason, velocity would be considered missing from the input stream and the algorithm would need to determine if the ship was a tanker using other data inputs, such as, hull number, silhouette, heat signature, etc.

Sparse Data (Ref 5), for these purposes, is considered when the system is working but no data is available. An example of sparse data might be a radar system not receiving any blips, in other words, the radar system is fully functional, but no tracks are present. Most times sparse data will be represented by a zero whereas a blank field represents missing data. For additional details regarding missing and sparse data refer to Ref. 1.

**DC - 17.** Analyze the Datasets for labeling bias including how the labeling process prevents mislabeling of data e.g., human, or labeling tool errors.

Labeling bias in ML is a type of error in which certain elements of a dataset are more heavily weighted and/or wrongly represented due to human, data, or tool errors. A biased dataset will result in a model that produces skewed outcomes, low accuracy levels and analytical errors. Sources of this bias include observer bias (confirmation bias) where an observer sees what they want to see in the data, consciously or unconsciously; or

measurement bias where for example, the training data is collected using one camera but using a different camera to capture operational data. It is critical to document how data is labelled and demonstrate consistency across different labelers (if human) or how bias is avoided if labelling is automated. There are inherent biases so there may be a need to have reviews or checks and balances on the data labelling.

**DC - 18.** Document process to monitor whether original Datasets is still representative of operational environment. Document any time or domain limitations for the applicability of the training data (i.e., specific missions, locales, or adversary).

The original training data set may have become stale and not be representative of the current operational environment. This issue is often referred to as ODD drift; when the domain of data the model is trained on is different from the domain the model encounters in deployment operationally. A process is necessary to collect data just prior to fielding and compare that data to the original training data to determine adequacy. This process should also include reverification frequency, how the comparison will be conducted, what performance indicators will be used to determine if the model is no longer fit for deployment.

Developers may want to include a means to measure or record field data so that it can be periodically assessed against the training data sets to ensure the trained model is still valid for the intended application.

**DC - 19.** Perform independent review on the Datasets.

Due to importance of the Datasets to the successful development of a model, for safety critical functions, independent validators shall peer review the development process to ensure adequacy of the training data. It is recommended that there is an independent validator to determine if the data set is sufficient for the model e.g., peer review of the development process and verifying the quality and quantity of the training data set.

**DC - 20.** Ensure any risks identified with the data that is used in the training, testing and validation of the ML function are presented at design reviews for early decision making, mitigation, and user awareness.

For safety critical functions that are assigned as SFCI 1 and SFCI 2, the data used during Training, Test and Validation should be presented at design reviews to ensure the selection and curation of data for safety critical functions are justified, rationale documented, and that there is leadership understanding and acknowledgment.

**DC - 21.** Provide data integrity process.

This task is an area of overlap between safety and security. Since the training data drives the composition of the algorithm during training, it is important that the creation and subsequent integrity of the data, part of which may become the T&E test set, has strong oversight. A possible tool to provide this oversight, especially when the data is coming from many, diverse sources with multiple touch points, until it reaches its destination of storage as a training set, is a technology called blockchain (Ref. 7). Other considerations include CRCs or Checksums depending on the criticality of the Function. The more critical the function, the more rigorous the method that should be applied. Characterize the failure modes and risks associated with data integrity.

**DC - 22.** Perform all LOR activities on Datasets if Datasets contain newly acquired data prior to utilizing Datasets to re-train the model.

New data that is included into (or data that was deleted from) the Datasets will likely change the behavior of the algorithm. Therefore, all LOR tasks identified in this section must be performed again when there is new training data to ensure no new hazards or causal factors are inadvertently introduced. While not changing any software code, changing the data could have a profound impact on the model and thus system functionality. Ensure newly collected data undergoes the same level of scrutiny as the original data prior to being utilized. Ensure that integration of new data into Datasets does not introduce issues.

**DC - 23.** Verify that there is no data leakage i.e., Datasets has information that is not representative of the ODD.

Data leakage is when information outside the aligned Datasets is used to train the model which will result in incorrect behavior of the algorithm. Measures must be put into place to ensure Datasets are protected from inadvertent inclusion of irrelevant data.

## E. ALGORITHM TASKS

In ML, an algorithm is a set of instructions or a procedure that is used to learn patterns from data and make predictions or decisions based on those patterns. An algorithm is a general recipe for how to perform a particular task, such as classification or regression.

A model, on the other hand, is a specific implementation of an algorithm that has been trained on a specific dataset to make predictions or decisions about new data. In other words, a model is the result of applying an algorithm to a dataset and "learning" the patterns in that data. The model represents the relationships between the input data and the output predictions that the algorithm has learned from the training data.

To create a model, you first need to choose an appropriate algorithm that is suitable for the task you are trying to perform. Once you have chosen an algorithm, you then train it on a dataset to learn the patterns in the data and create the model. The model can then be used to make predictions or decisions on new, unseen data.

Because of the complexity of ML models and their uncertainty in their output, extra rigor must be taken when using ML to support the execution of safety significant functions. When ML models are used to implement safety-significant functionality in a system, it is important that the following tasks ensure integrity in the ML development process.

**AL - 1.** For the algorithms considered, identify, and review best practices/existing standards (industry approaches specific to selected algorithms) which determines the chosen algorithm’s suitability to the implementation e.g., regarding bias, variance and “sweet spot” (point where algorithm has a high True Positive rate and a low False Positive rate).

Document in the ML Development Plan how algorithms were considered to determine if they were good candidates for the intended application e.g., based on prior utilization of the algorithm in similar situations.

**AL - 2.** Document the process that will be used to select the algorithm including measures and criteria that will be used in the selection process.

Ensure the correct algorithm is chosen for the intended use (level of complexity, transparency, processing time, performance/accuracy, stability, type of ML, safety case, explainability/interpretability, etc.). The process should include evaluation of why this algorithm is a better solution than a different algorithm or traditional software. Are multiple algorithms being used together to accomplish a function or provide safety redundancy? Ensure the correct intended use is considered. Review justification for the algorithm's design through measures of improved performance in terms of confusion Matrix parameters, e.g., sensitivity, specificity, precision.

**AL - 3.** Document the process and rationale for selection of the model architecture.

In ML, the term "model architecture" refers to the structure and design of ML algorithm. A model architecture defines how the data is processed by the algorithm, and how the algorithm learns to make predictions or decisions based on that data.

The architecture of a model typically consists of several layers, each of which performs a specific task in the data processing pipeline. These layers may include input layers, hidden layers, and output layers, each with their own set of parameters and functions. For example, in a neural network model, the input layer receives the raw data, which is then processed through a series of hidden layers that apply nonlinear transformations to the data. The output layer then produces the final predictions or decisions based on the transformed data. The specific architecture of a ML model can vary widely depending on the problem being solved.

**AL - 4.** Document how the cost function will be selected and used for the optimization of the ML function.

A cost function, also known as a loss function, is a mathematical function that measures the difference between the predicted output and the actual output of a model. It is used to evaluate the performance of the model during training and to guide the optimization process of the model's parameters. The goal of a cost function is to minimize the error between the predicted output and the actual output of the model. This error is often referred to as the loss or the cost. The cost function is a mathematical function that takes the predicted output and the actual output as inputs and outputs a scalar value that represents the cost or loss.

During training, the goal is to find a set of model parameters that minimize the cost function. This is typically done using an optimization algorithm, such as stochastic gradient descent (SGD) that iteratively updates the parameters in the direction of the steepest descent of the cost function.

The choice of cost function is important because it affects the behavior of the optimization algorithm and the performance of the model. A good cost function should be differentiable and continuous, and it should provide a meaningful measure of the performance of the model. The cost function should also be chosen to match the task at hand, such as classification or regression, and the type of output being predicted, such as continuous or discrete.

**AL - 5.** Document the training curriculum of the algorithm - document how the model was trained using the data set.

Ensure all variables in the training curriculum are identified such as order of training data (different order of training data may create a different model), timing of data, types of noise present in the training data, etc. Document the specifics of the training process (to include equipment used) so that any unintended influence can be identified. Note any variables in the training curriculum that were chosen with the intention to simulate the real-world conditions (environment).

**AL - 6.** Document process to update algorithm or retrain algorithm with new data, e.g., running the previous version of the algorithm in parallel to allow crosschecking.

An algorithm may need to be updated for various reasons, for example an algorithm to translate language may need retraining with new data for use with a certain dialect. Examples of updates to an algorithm include retraining, transfer learning and maintenance. In the same way that the initial training curriculum needs to be documented, the process to update/retrain an algorithm also needs to be documented. Include in the documentation any lessons learned from the need to update/retrain the algorithm.

**AL - 7.** Document how hyperparameters in the algorithm(s) selected were optimized.

Hyperparameters are parameters that are not learned from the training data but are set manually by the ML practitioner to control the behavior of the learning algorithm. These parameters are not learned by the algorithm during training but can have a significant impact on the performance of the model.

Hyperparameters are different from model parameters, which are learned by the algorithm during training to capture the underlying patterns in the data. Model parameters are typically adjusted through a process called backpropagation, in which the error between the predicted and actual output is used to adjust the weights of the neural network. Hyperparameters, on the other hand, are set by the practitioner before the training process begins and are used to control the behavior of the algorithm during training.

Choosing suitable hyperparameters and model architecture plays a crucial role in the success of the neural network architecture. Common hyperparameters to consider are learning rate, batch size, number of hidden layers, regulation strength, momentum, and weight decay.

Any hyperparameters that affect the safe behavior of the model safe should be identified/described and the rationale for choosing those hyperparameters should be documented. What method did the developer use to adjust the hyperparameters until the desired results were obtained? Would using a different method to adjust the hyperparameters (such as adjusting one parameter at a time versus making an adjustment to multiple parameters at once) result in a different optimization? Ensure any requirements for tools used for hyperparameter optimization are documented.

## F. CODING PHASE TASKS

Modern software development (coding) is multi-faceted. Often, software (including AI/ML development software) is a conglomeration of different code libraries and often relies on numerous sources to achieve the intended function of the software. ML, specifically, frequently takes advantage of Python packages and frameworks such as PyTorch or Google's TensorFlow in development of the model for their easily implementable toolsets and data processing capabilities. While the use of common code libraries is commonplace in ML development, the selection of how, when, and where to utilize these external resources must be conscientious and responsible. The use of libraries in deployed software to enable the ML capability must be analyzed carefully for impacts to the software behavior and failure modes. When used in safety critical applications ML development software must meet current coding development standards.

**CA - 1.** Document review of code that implements the ML relevant portions of all tools (e.g., data collection, data curation, training of the algorithm(s), automated testing).

ML development frequently requires multiple inputs and ongoing processes that are separate from strictly developing ML functionality. An example of this is software used for labelling data used to train models. Because the output of the labelling software produces an input for an ML project, the labelling software must be authorized and reviewed. In a similar sense, any software used to enable ML development requires a structured review i.e., Software Safety LOR as defined in the Joint Services Software Safety Engineering Handbook (JSSSEH) Implementation Guide. This is not only required of software packages, as seen in CA-2, but of software that is secondary to the ML development (e.g., test and evaluation software suite). See the Data Curation LOR Tasks for further details on data-related analysis.

Microsoft's Responsible AI Toolkit is an example of a resource that provides ML developers with insights to potential errors within their entire project (primary and

secondary software). The Responsible AI Toolkit also provides ML developers with diagnostic tools for addressing the issues and has debugging capabilities. While the use of this toolkit is not required (or feasible) for every project, similar capabilities, diagnostics, and testing mechanisms should be sought after.

JSSSEH Implementation Guide Section 3.9 Process Task 9.0: Perform Code-Level Safety Analysis Tasks for AI relevant portion of all tools would be an applicable activity, along with other LOR the end system implementation. Those activities should be reviewed and adhered to for these additional tools.

NOTE: For SFCI 1 and SFCI 2, the AI Developer supporting the review should be considered independent of the producing team/organization.

**CA - 2.** Determine that libraries and functions that are called within the code are robust and are of sufficient pedigree. Note that libraries and functions used within safety critical code are assessed as safety critical as well.

Before using code packages and frameworks from outside sources, establish a written purpose for the development task or sub-task (e.g., ML model validation). Having well defined levels of acceptable flexibility and risk acceptance criteria is a core component in determining what libraries, if any, are acceptable for use (see the Algorithm Development LOR Tasks for more details).

After determining that external packages are necessary and/or worth the risk trade-off, consider the credibility and reputation of the sources providing the functionality. Python Package Index (pypi.org) is an example of resources for discovering information about Python code packages. Other programming languages have comparable indexes for their packages. When determining the quality of a package, a few attributes to consider are authorship, licensing, release schedules, county of origin, code coverage testing, project funding, etc., Reference 4 further details a similar process and provides suggestions for evaluating the quality and necessity of packages. The paper evaluates packages from the R programming language, but the content is generically applicable.

Upon integrating a package into a larger code-base or project, validate that the package is working as intended and expected. When using multiple external packages,

interoperability may be a concern due to software configuration requirements. Perform multiple test and validation methods (e.g., integration testing, regression testing, simple validation) to ensure that the package does not have immediate consequences to the software. Isolating packages as much as possible is a best practice. If the project only requires the math.pi variable from Python's math standard library, instead of importing the entire python math library (import math) only import the functionality and variables you need and plan to use (from math import pi). The concern here is unused and untested code and potential unintended consequences.

Finally, all programs are different and have various requirements. With that, some techniques proposed above may work well for some groups while being obsolete for others. For example, some programs may have restrictions that prevent them from using outside sources for various reasons. If this is the case, programs developing their own libraries is an option. While developing all code libraries "in-house" might reduce risk in some respects, it may raise reliability concerns.

NOTE: For SFCI 1 and SFCI 2, the AI Developer supporting the review should be considered independent of the producing team/organization.

## G. T&E PHASE TASKS

T&E is an important assessment capability that feeds into the overall safety analysis, regardless of the type of system being developed. There are currently many different methods and procedures for executing the T&E process, each with different benefits and goals. When considering ML technologies, there may be different methods or procedures required to reach a satisfactory level of assurance. This would be determined by the technology developed, the application within the system, the overall concept of operations, and Safety Functional Criticality Index. These LOR activities provide additional recommended or suggested activities to increase the level of assurance in the system. Be advised that any tool used to generate/curate/test data shall have the same LOR as the function it performs IAW RA-5.

NOTE: For SFCI 1 and SFCI 2, the T&E should be independent from the AI Developer.

NOTE: For SFCI 1 and SFCI 2, the AI Developer supporting the review should be considered independent of the producing organization.

**T&E - 1.** Define Test and Validation process for the ML function to ensure appropriately selected T&E data is available in sufficient quantity for testing, and validation.

This element is comprised of two primary concerns: ensuring that there is enough data to use for testing and validation; and ensuring that the right breadth of data has been selected. This requires coordination between the Test and Evaluation engineer, Safety engineer, Data Scientist, and AI Developer.

To start, there should always be enough data available to perform the testing. As with non-ML testing activities, a few tests will likely be insufficient to obtain the needed results and assurance. However, just as important as quantity is quality, or at least ensuring that the right data has been selected. DC-2 looks to quantify the sourcing of data from which the test data will be selected. There are a variety of means to select data, but random selection may not be the most prudent. Random data should not be used for training, and instead withheld for use in testing only. The exception to this is K-Fold variation, which uses all data for training/testing, to predict model pedigree once trained appropriately.

This activity includes:

1 - Rationale for the method used to generate the subset of training, validation, and/or testing data.

2 - Percentage breakdown of the training to validation/testing data.

3 - Analysis on the diversity of data to cover different behaviors, operations, and environments.

4 - Considerations for off-nominal situations/scenarios

5 - Impact on safety of the data selection, to include identification of any safety specific data requirements, needs, or requests.

There is no specific method to address all these areas, and there may be additional unique challenges to specific sets of data. Conversations and collaboration with the Test and Evaluation engineer, Safety engineer, Data Scientist, and AI Developer should take

place to identify the needs (examples above) and which of these needs can be supported by the system process or need refinement.

**T&E - 2.** Document the test and validation methods and criteria.

The appropriate test and validation methods and criteria should be documented in the Test and Evaluation Master Plan (TEMP) or the Test Evaluation Strategy (TES). This documentation should consider the novel aspects of the assessment, such as the sourcing and selection of data and the utilization of Human System Integrate (HSI)/Human Machine Teaming (HMT). These methods and criteria will likely be unique and specific for a given data modality or algorithm and may need adjustment or refinement. However, standard best practices for documenting the methods and criteria would remain, but there may be additional concepts to add to that documentation. Example concepts and points to consider:

1 - Metrics and measures are used to confirm performance, function, and sufficiency of the ML function.

2 - Instrumentation requirements

3 - Test resource requirements

4 - Operational conditions and limitations

5 - Methods and criteria for testing input boundaries if the input state space is significantly large (e.g., computer vision applications)

6 - Determination of possible robustness and tolerance

7 - Determination of possible underfit or overfit

8 - Determination of methods and means to verify the written requirements to an appropriate level of confidence.

9 - If used, methods for Confusion Matrix, Receiver Operator Characteristic, and/or Precision Recall Plot

10 - Any additional methods or criteria as needed by any other applicable LOR activities.

**T&E - 3.** Develop detailed test plans that include specific behavior that should be checked during T&E

The TEMP or the TES should contain existing best practices focused on T&E. This documentation should consider the new aspects of the test assessment, such as the sourcing and selection of test and validation data. Test plans should contain traceability to training data and expected behavior.

Example plans, references, sources of information, and discussion points across the system lifecycle that may include information relevant to test cases:

1 - Data Safety Management Plan/Data Management Plan (or similar)

2 - Human-Machine Teaming Verification Plan (or similar)

3 - System Safety Working Groups

4 - Upfront integration of test needs and instrumentation in development (e.g., test hooks)

5 - Assurance and use cases.

6 - Transfer learning verification and validation

7 - Process for adversarial testing, if applicable

8 - Expected behavior of the system, as well as plans for assessing, understanding, and controlling emergent or unexpected behavior

9 - Any additional plans, references, or information as needed by any other applicable LOR activities.

**T&E - 4.** Conduct trials in an operationally representative simulated environment with Hardware in the Loop (HITL) to ensure there are no unforeseen behaviors observed in the ML function to preclude subsequent safety-critical testing.

The intent of HITL testing is to build incremental confidence in the safety and performance of the software, beyond what a simulated environment offers. A software test plan should be created which identifies the ML-related HITL test cases. This may be a distinct section of the test plan which addresses simulated environments or a distinct document. Most of the same guidelines with respect to simulated testing apply to HITL testing. Safety-related test cases should be identifiable and traceable to the software requirements. Test cases should test both nominal and off-nominal conditions. If possible, the system should be instrumented to give testers real-time insights into the ML algorithm.

Test cases should be assessed for sufficiency prior to test execution. There should be a Test Readiness Review (TRR) with clear entry and exit criteria. Software versions of all test items should be documented prior to test initiation and the test environment should be well-defined. All test anomalies should be identified and documented in a formal problem reporting system. Test anomalies should be assessed for safety implications. Test anomalies should also be assessed for being ML-related or not. If a safety related test anomaly has impact on real world environment testing, it may require correction and verification prior to initiating those future test efforts or external mitigations may be required.

**T&E - 5.** Conduct trials in an operationally representative real-world environment to ensure there is no unforeseen behavior observed in the ML function to preclude subsequent safety-critical testing.

A system-level software test plan should be created which identifies the ML-related test cases. Additionally, safety related test cases should be identifiable and traceable to the software requirements. Range/test site safety should be considered in test design, and a Test Analysis Working Group (TAWG) should be constituted to assure proper system limits (e.g., hard stops, manual overrides, power control) for the test environment and system are implemented. Test cases should test both nominal and off-nominal conditions. Test cases should be assessed for sufficiency prior to test execution. There should be a TRR with clear entry and exit criteria. Software versions of all test items should be documented prior to test initiation and the test environment should be well-defined. All test anomalies should be identified and documented in a formal problem reporting system. Test anomalies should be assessed for safety implications. Test anomalies should also be assessed for being ML-related or not. If the system performs in an unsafe manner during system-level testing, then a real-time assessment may be required to determine if system safety controls were sufficient to prevent a mishap.

**T&E - 6.** Determine unique range hazardous conditions when testing ML systems.

Test range safety requirements often require human intervention or other unique capabilities specifically for the range. Thus, special consideration must be given to range safety when the ML system is designed or when it is prepared for testing. These may involve independent flight/test termination systems or added operator intervention capability.

**T&E - 7.** Operational testing must be uniquely designed to properly test ML systems.

Operational testing must take into consideration how the CONOPS, tactics and procedures, and operational environment will change when the ML system is deployed. These changes must be incorporated into the design of the operational testing.

During operational testing, ML systems may require additional instrumentation to obtain necessary performance measures. Operational testing may require additional safeguards such as human in the loop, watchdog systems, or other means to ensure failures of the ML system do not propagate to mishaps in test.

## H. HUMAN-MACHINE TEAMING/HUMAN SYSTEM INTEGRATION TASKS

HMT and HSI are critical factors for system safety. The human operator may provide the ability to intervene or prevent a mishap from occurring due to a poorly designed and development system. However, by that same token, the human operator can also contribute to a mishap in an otherwise safe system. To help mitigate these risks, an understanding of the operator interaction with the system, the perceived notion of system capabilities, training of the operator, and level of confidence/trust by the operator (justified confidence/justified trust) should be measured and assessed. Each of the suggested LOR tasks for HMT/HSI are related and should be reviewed together during assessments where appropriate. Many of these HMT LOR tasks will reference DODI 5000.95 – Human Systems Integration in Defense Acquisition, of which familiarity would be beneficial. Accompanying military standards MIL-STD-1472 and MIL-STD-46855 may be of use, as directed by the HSI engineer.

**HMT - 1.** Document how the operator has been educated, trained, and qualified to ensure sufficient understanding of ML capabilities and limitations to prevent over or under confidence in the ML function.

The human operator is a critical piece in any system deployment. If an operator does not fully understand and comprehend the system's capabilities and limitations, that can lead to system safety risks. One of the ways to ensure that an operator understands the system is through education and training to properly level set the operator. The feedback from training assessment can be used to determine the confidence of the operator, and the ability of the operator to understand the appropriate application of the system.

In human systems engineering, the Personnel and Training domains as outlined in DODI 5000.95, Human Systems Integration in Defense Acquisition, can assist with determining the proper military occupational specialties, knowledge, skills, and abilities. Coordination with training experts and human system integration engineers will be needed to fully assess this activity. Examples of meeting this need:

1 - Maintaining a training plan that includes the full capabilities and limitations of the system.

2 - Ensuring users are qualified and trained for the given requirements and needs for using/interacting with the ML in the system usage and operational environment.

3 - Determination of level of involvement of the operator during system development.

4 - M&S with operator involvement to better understand how the human and machine team together, and the impacts to that teaming (positive and negative).

Method for operators to provide feedback on system use during deployment.

Ensuring that training and guidance is very clear about the capabilities and limitations of the system, and the effectiveness of that training will give assurance that the operator can understand the intent and use of the system. The training should stress the capabilities and limitations to the operator so they can fully understand when the system is not appropriate for a mission or not providing the correct information, thus reducing the

confidence of the operator in the ML system to an appropriate level for the current operation. Additionally, safety mitigations that require operator involvement should be traced to specific operator training and should include verification to confirm they are in place and correct.

**HMT - 2.** Document analysis of human machine interaction with the ML function.

Part of the human system integration is an understanding of how the operator interacts with the system, and how information is presented to the operator. Given the possible safety critical nature of the system, the information provided can be vital for critical decision making by the operator. There should be a balance of providing the right amount of information to allow for informed decisions, but not an amount that will overwhelm the operator and prevent or hinder a decision. A human may be relied upon to confirm the output of the algorithm before the next step occurs, which, for example, may be firing a weapon or identifying a target. On the reverse side, it may be critical for the human to be able to direct the machine not to fire or not to target an object. The Human Factors Engineering domain in HSI covers the principles to ensure that system design considerations are compatible with the operator's capabilities and limitations. By having that understanding, it is less likely for an operator to have issues interpreting data and making decisions. Example of areas of concern that should be addressed:

1 - Design of the interface for operator and maintainer.

2 - Accessibility

3 - Avoid the need for extensive/complicated training.

4 - Avoid the need for excessive cognitive, physical, and sensory skills. Compare these impacts against traditional software solutions.

5 - More details can be found in MIL-STD-1472 and MIL-STD-46855

An HSI engineer can provide this analysis and any recommendations during design and prototyping of the system. This may include monitoring of an operator during prototyping or potential system integrations to gain feedback and analysis on activities.

The activities and artifacts from the analysis conducted in HMT-1 will help to determine the results of this LOR analysis, especially in the traceability to the ML functions specified.

**HMT - 3.** Analyze and assess any trust processes or documentation to identify mitigations of hazardous conditions or causes of hazardous conditions.

For each ML-enabled operation requiring HMT, the operation will be assessed and categorized for its relationship to safety. Each operation will be categorized as necessary for hazard mitigation, being a potential hazard cause or not impacting safety. For operations that serve as hazard mitigations, operator confidence in executing the operation will be analyzed and assessed. It will be determined if the mitigation is singular or if multiple mitigating factors exist to prevent the hazard from occurring. The confidence in ML-enabled HMT operations will be assessed through test trials. Each ML-enabled operation will be considered separately.

For ML-enabled HMT operations that can be causal factors for hazards, a similar process will be followed. For each operation that acts as a causal factor, mitigations will be identified and characterized for effectiveness. Again, the confidence in ML-enabled HMT operations that act as hazard causal factors will be assessed through test trials. Each ML-enabled operation will be considered separately.

These assessments can take place as part of the analysis under T&E-3 through T&E-6, HMT-1, HMT-2, to ensure that the operator interaction is also analyzed during testing.

**HMT - 4.** Conduct testing of HMT tasks relative to system safety.

HMT tasks of ML functionality should be integrated into the overall system testing approach. Design of test cases of HMT functions should be traceable to system level hazards, specifically to address causes where HMT shortfalls could result in a hazardous outcome. There will be sufficient test cases to ensure that each HMT causal factor is identified in the hazard analyses. Measurements should be applied to test case execution to document if the execution is both timely and accurate. Excessive time and inaccurate decisions will be considered in the likelihood of a given hazard. It will be documented that

the testers are sufficiently trained in the use of the system and the ML prior to test execution.

These assessments can take place as part of the analysis under T&E-3 through T&E-6, HMT-1, HMT-2, to ensure that the operator interaction is also analyzed during testing.

**References**:

1 – Hawkins, R., Paterson, C., Picardi, C., Jia, Y., Calinescu, R., Habli, I. (2021), Guidance *on the Assurance of Machine Learning in Autonomous Systems (AMLAS) v 1.1*

2 – *EASA Concept Paper: First Usable Guidance for Level 1 Machine Learning Applications.* (2021). European Unition Aviation Safety Agency.

3 –*Army Machine Learning Certification Supplement* (AMLCS). (202X). [Draft under internal review], U.S. Army Combat Capabilities Development Command Aviation & Missile Center, Redstone Arsenal, AL 35898.

4 – Kandula, Sudheer & Sree Ranga Vasudha Moda, *Hardware Strategies for Network Optimization Supporting AI Workloads.* (2023). International Research Journal of Modernization in Engineering Technology and Science, Vol 05/Iss10/Oct2023.

5 – Nagy, Bruce. (2022). NAWCWD TP 8864 - *Level of Rigor for Artificial Intelligence Development*. Naval Air Warfare Center Weapons Division, China Lake, CA.

6 – Nagy, Bruce (2021). *Increasing Confidence in Machine Learned (ML) Functional Behavior during Artificial Intelligence (AI) Development using Training Data Set Measurements,* Proceedings of the Eighteenth Annual Acquisition Research Symposium, Naval Postgraduate School.

7 – Kendal, Anthony, Das, Arijit, Nagy, Bruce, Ghosh, Avantika, (2021). *Blockchain Data Management Benefits by Increasing Confidence in Datasets Supporting Artificial Intelligence (AI) and Analytical Tools using Supply Chain Examples*, Proceedings of the Eighteenth Annual Acquisition Research Symposium, Naval Postgraduate School.

8 – Wendt CJ, Anderson GB (2022). Ten Simple Rules for Finding and Selecting R packages. PLoS Comput Biol 18(3): e1009884. https://doi.org/10.1371/journal.pcbi.1009884

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX B - LEVEL OF RIGOR MATRIX

NOTE: It is the authors' assessment that SFCI 1 Machine Learning (ML) application cannot be employed at an acceptable risk level with LOR activities alone. The authors highly recommend employing interlocks, or guardrails, to reduce the autonomy of the ML functions and performing these or other appropriate LOR activities, such as software safety LOR, for verification and assurance of the mitigations.

Table 1 - Requirements Analysis Objectives

| | Level or Rigor (LOR) Activity | Primary Responsibility | Support Responsibility | Baseline | Review | Safety Function Criticality Index (SFCI) | | | Representative Artifacts Produced |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Requirements Analysis** | | | | | | | | |
| RA - 1 | Assess and document justification for the proposed use of ML over operator input, more traditional software, firmware, or hardware technique. Provide rationale as to why an ML solution is a "better" fit to the functional requirement. | System Engineer | AI/ML System Developer/IPT | PR | SFR/PDR | | | | Analysis report |
| RA - 2 | Document justification (selection criteria) of selected supervised ML technology or algorithm/model (Naive Bayes, random forest, etc.) that is the most appropriate for use in this function e.g. would this algorithm provide for the best operational performance. | AI/ML System Developer | System Engineer | PR | SFR/PDR | | | | Analysis report |
| RA - 3 | Determine how performance for this ML function is measured e.g. success rate (confidence % or similar) . | AI/ML System Developer | System Engineer | PR | PDR | | | | Analysis report |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| RA - 4 | Document the Change Management process for the data (Training, Test and Validation), algorithm(s), model, and associated tools. | AI/ML Developer | System Engineer, CM SME | PR | SFR/PDR | | | | ML Development Plan, Configuration Management Plan |
| RA - 5 | Ensure ML development environment and tools are addressed in the Software Develop Plan and the appropriate Level of Rigor is applied | AI/ML Developer Software Engineer | System Engineer, | PR | SFR/PDR | | | | Software Development Plan |
| RA - 6 | Ensure that the System Safety Program Plan includes sections for ML System Safety | AI/ML Developer Software Engineer | System Engineer, | PR | SFR/PDR | | | | Software Development Plan |

Table 2 - Architecture Analysis Objectives

| | Level or Rigor (LOR) Activity | Primary Responsibility | Support Responsibility | Baseline | Review | Safety Function Criticality Index (SFCI) | | | Representative Artifacts Produced |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Architecture Analysis** | | | | | | | | |
| AA - 1 | Document ML training data source (or sensor) architecture e.g. modality type. | AI/ML Developer | System Engineer | PR | CDR | | | | ML Development Plan (Architectural Description), ML Data Management Plan |
| AA - 2 | Justify the Function Criticality Index (FCI) of the ML function. Determine if other control entities (such as a human operator, software, firmware, hardware) can be inserted into the loop to reduce the Function Control Category (FCC). | AI/ML Developer | System Engineer | | CDR | | | R | Functional Hazard Analysis (FHA), ML Development Plan (Architectural Description) |

Table 3 - Design Analysis Objectives

| | Level or Rigor (LOR) Activity | Primary Responsibility | Support Responsibility | Baseline | Review | Safety Function Criticality Index (SFCI) | | | Representative Artifacts Produced |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Design Analysis** | | | | | | | | |
| DA - 1 | Determine if the design includes quality check mechanisms to ensure the integrity of the input data, ensuring that corrupted or incorrectly formatted data is not provided to the Model. | AI/ML Developer | System Engineer, Verification Engineer | | PDR | | R | R | ML Development Plan, ML Verification Plan, System Design Document |
| DA - 2 | Document behavior of function when the algorithm experiences an operational environment that is beyond the limits of the training data. What does the function do when faced with input it is not trained for? | AI/ML Developer | System Engineer | | TRR | | R | R | ML Development Plan, System Design Document |
| DA - 3 | Determine if there are 'wrappers' around the function that are verifying and/or intervening when outputs of the AI function are unreasonable e.g. outside a defined range, intelligible. | AI/ML Developer System Engineer | Software Engineer | | PDR | | | R | ML Development Plan, System Design Document |
| DA - 4 | Document how the API/SQL/MSG interface is verified | AI/ML Developer | Software Engineer | | PDR | | R | R | ML System Verification Plan |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DA - 5 | Document how databases housing all data (training, test, validation, and operational databases) are protected from unintentional corruption e.g. database physical or electronic damage. | AI/ML Developer, Data Analytics Engineer | System Engineer, Configuration Management Engineer, Cyber Engineer | | PDR | | R | R | Data Analytics Report, ML Development Plan |
| DA-6 | Document that wherever possible, the design makes use of predefined and well understood modules | AI/ML Developer, Data Analytics Engineer | System Engineer, Configuration Management Engineer, Cyber Engineer | | PDR | | R | R | Data Analytics Report, ML Development Plan |
| DA-7 | Document that wherever possible, the design makes use of predefined and well understood architectures | AI/ML Developer, Data Analytics Engineer | System Engineer, Configuration Management Engineer, Cyber Engineer | | PDR | | R | R | Data Analytics Report, ML Development Plan |
| DA - 8 | Design for interoperability with human operators, human co-warfighters, and other systems | AI/ML Developer, Data Analytics Engineer | System Engineer, Configuration Management Engineer, Cyber Engineer | | CDR | | R | R | Data Analytics Report, ML Development Plan |

Table 4 – Data Curation Objectives

| | Level or Rigor (LOR) Activity | Primary Responsibility | Support Responsibility | Baseline | Review | Safety Function Criticality Index (SFCI) | | | Representative Artifacts Produced |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Data Curation** | | | | | | | | |
| DC - 1 | Identify available sources of data and determine what type of data (real or synthetic) will be used to train, test, and validate the ML function. | AI/ML Developer, Data Analytics Engineer | | | SRR | | R | R | Data Analytics Report, ML Development Plan |
| DC-2 | Ensure that training, testing, and validation data appropriately identified and separated from each other. | | | | PDR | | R | R | |
| DC - 3 | Document the type of data (real or synthetic) and sources/conditions/metadata associated with it, identify any shortcomings. | Data Analytics Engineer | AI/ML Developer | | PDR | | R | R | Data Analytics Report |
| DC - 4 | Document how the data is prepared for the model, include how the data retains the desired characteristics for future training iterations. | Data Analytics Engineer | AI/ML Developer | | PDR | | R | R | Data Analytics Report |
| DC - 5 | If generating synthetic data, document the data generation process. | Data Analytics Engineer | AI/ML Developer | | PDR | | R | R | Data Analytics Report |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DC - 6 | Analyze and assess if training, test, and validation data (real or synthetic) reflects real world operations. | Data Analytics Engineer | System Safety Engineer, AI/ML Developer | | PDR | | R | R | Data Analytics Report |
| DC - 7 | Analyze and assess the attributes of the objective to determine whether the training, test, and validation data represents adequate quantity, quality, attributes for the desired classification. | Data Analytics Engineer | AI/ML Developer, System Safety Engineer | | PDR | | R | R | Data Analytics Report |
| DC - 8 | By ML class, define the rating of importance of attributes (precedence) that the ML algorithm needs in the training data to perform at the highest success rate based on the operational environment during deployment | Data Analytics Engineer | AI/ML Developer, System Safety Engineer | | PDR | | R | R | Data Analytics Report |
| DC - 9 | Once the priority of attributes has been determined above, verify that the attributes are appropriately reflected in the training data (real or synthetic). | Data Analytics Engineer | AI Developer, System Engineer | | PDR | | R | R | Data Analytics Report |
| DC - 10 | Have the appropriate features in the training data been selected to allow the ML function to perform correctly in the operational environment? | Data Analytics Engineer | AI/ML Developer, System Engineer | | PDR | | R | R | Traceability matrix in Data Analytics Report |
| DC - 11 | Assess any missing attributes or features with the operational community and the system safety working group | Data Analytics Engineer | AI/ML Developer, System Engineer | | PDR | | R | R | Data Analytics Report |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DC - 12 | Determine if the training, test, and validation data included instances of sufficient noise/clutter that allows the algorithm to function robustly when deployed. | Data Analytics Engineer | AI/ML Developer, System Safety Engineer | | PDR | | | R | Data Analytics Report |
| DC - 13 | Determine if the training, test, and validation data included sufficient operational complexity that allows the algorithm to function robustly when deployed. | Data Analytics Engineer | AI/ML Developer, System Safety Engineer | | PDR | | | R | Data Analytics Report |
| DC - 14 | Document process to determine if the training data (real or synthetic) will result in underfitting or overfitting of each Class. | Data Analytics Engineer | AI/ML Developer, System Engineer | | PDR | | R | R | Data Analytics Report |
| DC - 15 | Document traceability of training, test and validation data to operational conditions and operational requirements | Data Analytics Engineer | AI/ML Developer | | CDR | | R | R | Data Analytics Report |
| DC - 16 | Analyze the system and document if training, test, and validation data (synthetic or real) included scenarios where there is sparse, inconsistent, interrupted, or missing data (e.g. data with missing attributes). | Data Analytics Engineer | AI/ML Developer | | PDR | | | R | Data Analytics Report |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DC - 17 | Analyze the training, test, and validation data set for labeling bias including how the labeling process prevents mislabeling of data e.g. human or labeling tool errors. | Data Analytics Engineer | AI/ML Developer | | PDR | | | R | Data Analytics Report |
| DC - 18 | Document process to monitor whether original training data is still representative of operational environment, whether through periodic reverification of environment or target/threat data. Also, document if there is a time limitation for the applicability of the training data (i.e. specific missions, locales, or adversary). | Data Analytics Engineer | AI/ML Developer | | PDR | | | R | Data Analytics Report |
| DC - 19 | Perform independent review on the data set. | System Safety Engineer | Data Analytics Engineer | PR | PDR | | | | Data Analytics Report SSHA |
| DC - 20 | Ensure any risks identified with the data that is used in the training, testing and validation of the ML function is presented at design reviews for early decision making, mitigation, and user awareness | Data Analytics Engineer | AI/ML Developer, System Safety Engineer | PR | SRR/PDR/ CDR/TRR | | | | Data Analytics Report |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| DC - 21 | Provide data integrity process | Data Analytics Engineer | AI/ML Developer, System Safety Engineer | PR | PDR | | | | Data Analytics Report |
| DC - 22 | Perform all LOR activities on dataset if dataset contains newly acquired data prior to utilizing dataset to re-train the model | | | PR | All | | | | |
| DC - 23 | Verify that there is no data leakage i.e. training dataset has information that is contrary the operational environment. | Data Analytics Engineer | AI/ML Developer | | PDR | | R | R | Data Analytics Report |

Table 5 – Algorithm Development Objectives

| | **Level or Rigor (LOR) Activity** | **Primary Responsibility** | **Support Responsibility** | **Baseline** | **Review** | **Safety Function Criticality Index (SFCI)** | | | **Representative Artifacts Produced** |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Algorithm Development** | | | | | | | | |
| AL - 1 | For the algorithms considered, identify and review best practices/existing standards (industry approaches specific to selected algorithms) which determines the particular chosen algorithm's suitability to the implementation e.g. regarding bias, variance and "sweet spot" (point where algorithm has a high True Positive rate and a low False Positive rate). | AI/ML Developer | AI/ML Developer | PR | PDR | | | | ML Development Plan, list of best practices used |
| AL - 2 | Document the process that will be used to select the algorithm including measures and criteria that will be used in the selection process. | AI/ML Developer | AI/ML Developer | PR | PDR | | | | ML Development Plan |
| AL - 3 | Document the process that was used to select the model architecture and hyperparameters. | AI/ML Developer | Design Architect | PR | PDR | | | | ML Development Plan |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| AL - 4 | Document how the cost function will be selected and used for the optimization of the ML function. | AI/ML Developer | Technology SME Developer Software Design Architect | PR | PDR | | | | ML Development Plan |
| AL - 5 | Document how the success rate of the model is measured. | AI/ML Developer | Technology SME Developer Software Design Architect | PR | PDR | | | | ML Development Plan |
| AL - 6 | Document how hyper-parameters in the algorithm(s) selected will be optimized. | AI/ML Developer | Technology SME Developer Software Design Architect | PR | PDR | | | | ML Development Plan |
| AL - 7 | Document the training curriculum of the ML algorithm (different order of training data may create a different model) - document how the model will be trained using the data set | AI/ML Developer | Technology SME Developer Software Design Architect | PR | PDR | | | | ML Development Plan |
| AL - 8 | Document process to update algorithm or retraining of algorithm with new data, e.g. running the previous version of the algorithm in parallel to allow crosschecking. | AI/ML Developer | Technology SME Developer Software Design Architect | PR | PDR | | | | ML Development Plan |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| AL - 9 | What mitigation approach is being used for robustness/hardening of the algorithm with respect to operational corruption of model parameters? | AI/ML Developer | Developer<br>Software Safety<br>Data Scientist<br>Technology SME Developer<br>Software Design Architect | PR | PDR | | | | ML Development Plan |

Table 6 – Coding Objectives

| | **Level or Rigor (LOR) Activity** | **Primary Responsibility** | **Support Responsibility** | **Baseline** | **Review** | **Safety Function Criticality Index (SFCI)** | | | **Representative Artifacts Produced** |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Coding** | | | | | | | | |
| CA - 1 | Document review of code of all tools relevant to the development of the ML model (e.g., data collection, data curation, training of the algorithm(s), automated testing). | AI/ML Developer | System Safety Engineer, Software Engineer | PR | TRR (of the tool development pipeline) | | | | Code Review Report |
| CA - 2 | Determine that libraries and functions that are called within the code are robust and are of sufficient pedigree. Note that libraries and functions used within safety critical code are assessed as safety critical as well. | AI/ML Developer | System Safety Engineer, Software Engineer | PR | PDR | | | | ML/Software Development Plan |

Table 7 – Test and Evaluation Objectives

| | Level or Rigor (LOR) Activity | Primary Responsibility | Support Responsibility | Baseline | Review | Safety Function Criticality Index (SFCI) | | | Representative Artifacts Produced |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Test & Evaluation** | | | | | | | | |
| T&E - 1 | Document Test and Validation process for the ML function to ensure appropriately selected T&E data is available in sufficient quantity for testing. | T&E Engineer | AI/ML Developer | PR | CDR | | | | T&E Plan, Data Management Plan |
| T&E - 2 | Document the test and validation methods and criteria. | T&E Engineer | AI/ML Developer | PR | CDR | | | | T&E Plan |
| T&E - 3 | Develop detailed test plans that include specific behavior that should be checked during T&E. | T&E Engineer | Independent AI/ML Developer | PR | CDR | | | | T&E Plan and Test Descriptions/Cases |
| T&E - 4 | Conduct trials in an operationally representative simulated environment with Hardware in the Loop (HITL) to ensure there are no unforeseen behaviors observed in the ML function to preclude subsequent safety-critical testing. | T&E Engineer | AI/ML Developer, System Safety Engineer | PR | TRR | | | | T&E Results |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| T&E - 5 | Conduct trials in an operationally representative real-world environment to ensure there are no unforeseen behavior observed in the ML function to preclude subsequent safety-critical testing. | T&E Engineer | AI/ML Developer, System Safety Engineer | | TRR | | R | R | T&E Results |
| T&E - 6 | Range safety is a unique issue when testing AI/ML systems. | T&E Engineer | AI/ML Developer, System Safety Engineer | | TRR | | R | R | T&E Plans/Results |
| T&E - 7 | Operational testing must be uniquely designed to properly test ML systems | T&E Engineer | AI/ML Developer, System Safety Engineer | | TRR | | | R | T&E Results |

Table 8 – Human Machine Teaming Objectives

| | **Level or Rigor (LOR) Activity** | **Primary Responsibility** | **Support Responsibility** | **Baseline** | **Review** | **Safety Function Criticality Index (SFCI)** | | | **Representative Artifacts Produced** |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 4 | 3 | 2 | |
| | **Human Machine Teaming** | | | | | | | | |
| HMT - 1 | Document how the operator has been educated, trained, and qualified to ensure sufficient understanding of ML capabilities and limitations to prevent over or under confidence in the ML function. | Human Systems Integration Engineer | AI/ML Developer, System Safety Engineer, System Engineer | PR | TRR | | | | Human Systems Integration Analysis, Operating and Support Hazard Analysis (O&SHA) |
| HMT - 2 | Document analysis of human machine interaction with the ML function. | Human Systems Integration Engineer | System Safety Engineer, System Engineer | | CDR, TRR | | | R | Human Systems Integration Analysis, Operating and Support Hazard Analysis (O&SHA) |
| HMT - 3 | Analyze and assess any trust processes or documentation (confidence) to identify mitigations of hazardous conditions or causes of hazardous conditions | Human Systems Integration Engineer | System Safety Engineer, System Engineer | | PDR | | | R | System Hazard Analysis (SHA), finalize analysis in Operating and Support Hazard Analysis (O&SHA) |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| HMT - 4 | Conduct testing of HMT tasks relative to system safety. | Chief Engineer / Human Systems Integration Engineer | System Safety Engineer, System Engineer, T&E Engineer | | TRR | | | R | System Hazard Analysis (SHA), finalize analysis in Operating and Support Hazard Analysis (O&SHA), TEMP, T&E Results |
| HMT-5 | Train the operator and team members as if part of the system. | Chief Engineer / Human Systems Integration Engineer | System Safety Engineer, System Engineer, T&E Engineer | PR | TRR | | | | System Hazard Analysis (SHA), finalize analysis in Operating and Support Hazard Analysis (O&SHA), TEMP, T&E Results |
| HMT-6 | Be aware of the operator effect on the operation of the ML system. | Chief Engineer / Human Systems Integration Engineer | System Safety Engineer, System Engineer, T&E Engineer | PR | TRR | | | | System Hazard Analysis (SHA), finalize analysis in Operating and Support Hazard Analysis (O&SHA), TEMP, T&E Results |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| HMT-7 | Understand and clearly define operational and teaming suitability of the ML system | Chief Engineer / Human Systems Integration Engineer | System Safety Engineer, System Engineer, T&E Engineer | PR | TRR | | | | System Hazard Analysis (SHA), finalize analysis in Operating and Support Hazard Analysis (O&SHA), TEMP, T&E Results |

| **Legend:** |
|---|
| **PR: Prerequisite Requirement – Required regardless of LOR or required in order to assess and determine LOR**<br><br>**R: Required for assigned LOR** |