SYM-AM-21-094

EXCERPT FROM THE PROCEEDINGS OF THE EIGHTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

Functional Hazard Analysis and Subsystem Hazard Analysis of Artificial Intelligence/Machine Learning Functions Within a Sandbox Program

May 11–13, 2021

Published: May 10, 2021

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

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

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

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

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

Functional Hazard Analysis and Subsystem Hazard Analysis of Artificial Intelligence/Machine Learning Functions Within a Sandbox Program

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

Gunendran (Guna) Sivapragasam—earned his BS in electrical engineering from the University of Technology, Malaysia, and then received his MBA from the College of William & Mary in Virginia. Sivapragasam joined Naval Surface Warfare Center Dahlgren Division (NSWCDD) in August 2009 and is currently the Technology System Safety Lead for the System Safety Division (R40). In this role, he has been participating in and leading several efforts focused on developing system safety capability to do hazard analysis of new technologies being introduced in U.S. Navy weapon programs, including systems that employ AI. [gunendran.sivapragas@navy.mil]

Loren Edwards—is an Aerospace Engineer working for System Safety at Naval Air Warfare Center, Weapons Division, at China Lake. He graduated from Cal Poly San Luis Obispo in 2017 with a BS in aerospace engineering. He performed Computational Fluid Dynamics analysis work before moving into System Safety. He has attended classes on and performed Functional Hazard Analyses and supports multiple programs in different stages of development. He is working towards solving key safety questions involved with the use of Machine Learning and Artificial Intelligence to better understand and guarantee the safety of these evolving technologies. [loren.edwards@navy.mil]

Abstract

Development of advanced Artificial Intelligence (AI)/Machine Learning (ML) system-enabled weapons and combat systems for deployment in the U.S. Navy has become a reality. This is also true for the other armed forces, as well as in homeland security and even the Coast Guard. From the Navy standpoint, the Naval Ordnance Safety and Security Activity (NOSSA) is attempting to get ahead of the acquisition cycle by focusing on the development of policies, guidelines, tools, and techniques to assess mishap risk in Safety Significant Functions (SSF) that are identified. NOSSA's efforts have the potential of influencing the acquisition community, including in requirements, development, and test and evaluation engineering. This paper makes recommendations for the Functional Hazard Analysis (FHA) and Subsystem Hazard Analysis (SSHA) analysis templates and focuses on ways to decrease autonomy within system operations and increase its correlated Software Control Category (SCC). The questions and discussions devised from this research aim to form guidance and offer best practices to address AI/ML system safety issues.

Introduction

The Department of Defense (DoD) is rapidly approaching the point where system safety practitioners will need to conduct mishap risk assessments on AI functions within upgraded systems being deployed in the Fleet. These systems will be crucial to ensure the DoD retains its dominance in military power (Brose, 2020). The safety community will soon be required to conduct system safety analysis on systems, including weapon systems that contain Artificial Intelligence (AI)/Machine Learning (ML) functionality (National Defense Authorization Act [NDAA], 2021; National Security Commission on Artificial Intelligence [NSCAI], 2021). AI

functions present unique challenges to system safety practitioners to identify hazards, assess risk, and identify risk mitigation measures. This includes how to properly employ a system with AI capability in an operational or tactical environment while reducing the probability of a mishap. Currently, no guidance exists on how to conduct system safety analysis on AI/ML functions, and this will prevent the certification of these systems for deployment (Naval Sea Systems Command [NAVSEA], 2008; National Institute of Standards and Technology [NIST], 2019).

The Problem

Assuring safety in AI/ML systems is a considerable challenge to current safety processes for traditional software. Traditional software can be assessed for safety through code review, and traditional software outcomes can be analyzed through automated code analysis techniques such as Modified Condition/Decision Coverage (Joint Software Systems Safety Engineering Workgroup, 2017). Together, these and other methods can provide a rigorous understanding of how software will function in a given situation, assuring some desired level of safety. However, a developed and trained AI/ML system cannot be analyzed with current analysis methods, and though it is theoretically possible for some ML designs (and completely impossible for others) to exhaustively test all inputs and outputs of an AI/ML software function, the calculation time required makes even small systems almost impossible to analyze. These issues, combined with the unique challenge of unpredictable real-world corner cases, result in AI/ML functions having an inherent lack of safety due to unknown, unanalyzable, and untestable factors (Sodhani, 2018).

Within the DoD, MIL-STD-882E guides the software safety process. This standard provides a method for categorizing safety significant software based on its level of autonomy, called the Software Control Category (SCC). SCC 1, Autonomous, defines the highest level of autonomy, while SCC 4, Influential, defines the least autonomous category of safety significant software. These SCCs are combined with the severity of related hazards to define a Software Criticality Index (SwCI). Each SwCI level requires a requisite Level of Rigor (LOR), or a specific set of tasks to be completed before that safety significant software is considered “safe,” or representing a certain level of acceptable risk for the system. SwCI 1 requires the most effort to achieve LOR, while SwCI 4 requires the least amount of effort to achieve LOR.

For software where functional failure could lead to catastrophic hazards and that either has control over safety significant hardware or provides safety-critical information, the safest SCC possible is SCC 3, Redundant Fault Tolerant, which results in SwCI 2 (MIL-STD-882E; Defense Standardization Program Office, 2012). If this function were instead SCC 1, Autonomous, or SCC 2, Semi-Autonomous, the resulting SwCI would be 1. In addition to the SwCI 2 LOR tasks, SwCI 1 LOR tasking additionally requires code level analysis, such as including MC/DC or equivalent testing (JS-SSA, 2017). This means that if the Safety Significant Function (SSF) that could lead to a catastrophic hazard is an AI/ML function, it would likely be impossible to perform full LOR tasking on that function, creating a considerable gap in software safety.

The Need

Unless new analysis techniques are developed that can address the specific issues described previously, the most effective way to increase confidence of safe operations in AI/ML systems is to decrease the safety significance of AI software. Per MIL-STD-882E, this can be accomplished by lowering the potential mishap Severity or the SCC of the function. The SCC is used to define the level of control that software has over SSFs. The higher the number (from 1 to 5), the less safety impact the software has. For catastrophic hazards where the software either has control over safety significant hardware or provides safety-critical information, SCC 3 is the safest possible category and should be the goal for all traditional and AI software

functions. There is increased difficulty in reaching this SCC for AI/ML, however, due to the fact that in many applications of AI/ML, the AI system is independent (autonomous) and may be the only system that reviews data and makes decisions based on that data (Sodhani, 2018).

In addition to this, the many uses of AI/ML throughout government and industry do not follow defined procedures for guaranteeing safe operations. The current processes used to determine how safe AI/ML software is, and the processes used to decrease the risk of hazards due to or involving AI/ML, vary widely and are not consistent between companies and government agencies (NSCAI, 2019). These varied approaches not only result in inconsistency and lack of safety rigor in deployed systems, but they also decrease trust in AI/ML technology. To address both of these issues, consistent approaches to AI/ML system safety analysis must be developed.

In many modern implementations of AI/ML, there are neither components nor systems in place that actively decrease the autonomy level of these specially developed software functions.

Figure 1 describes the process for performing a Functional Hazard Analysis (FHA), which is the primary analysis used to determine SCC and SwCI determinations for safety significant software. In reviewing Figure 1, several questions are posed with regard to AI/ML:

- Are unique tools needed because of the presence of AI/ML to complete this analysis?
- How would we complete this determination for an AI/ML deployed system?
- Are current SwCI definitions appropriate for AI/ML?

These questions, alongside proposed answers and solutions to them, are presented in this paper.

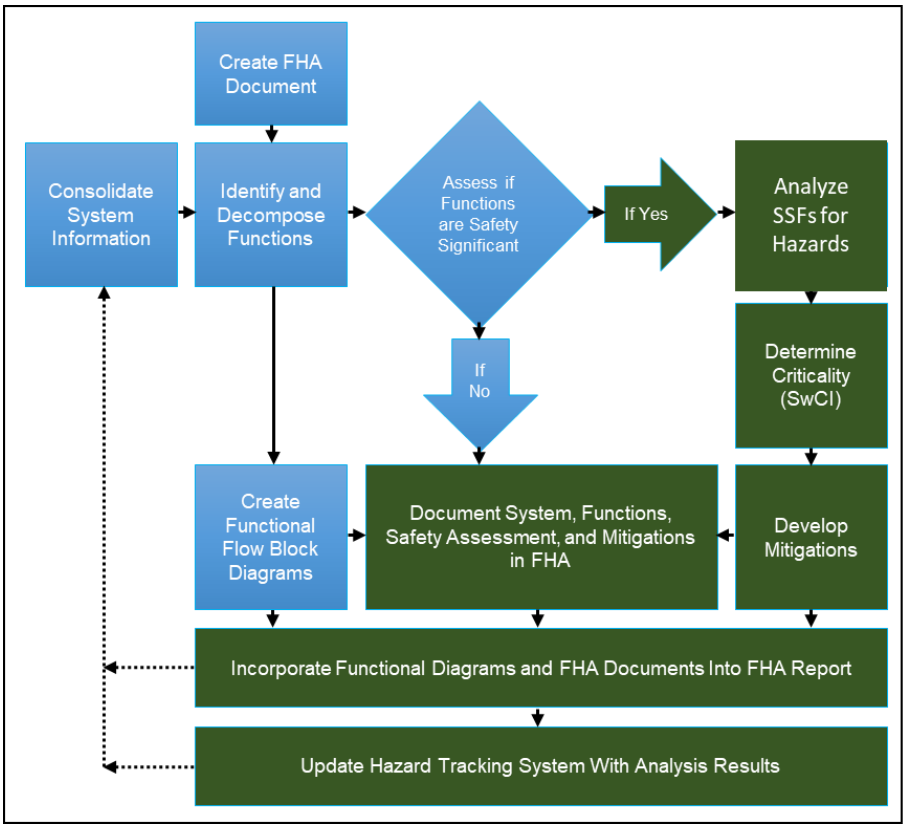


Figure 1. FHA Workflow

The Goal

This paper focuses on how to identify an AI safety critical function, gives recommendations to reduce the function's autonomy level, provides a format on how LOR for an AI/ML function can be identified, and includes some initial examples of AI/ML unique tasks for LOR. The goal of the research is to provide processes, questions, discussion points, and insights regarding the organization of the safety analysis in the form of tables and specifically labeled column headings. These structures and format recommendations are in support of system safety practitioners, providing guidance on how to conduct rigorous safety analysis on AI/ML software functions being deployed in weapon (MIL-STD-882E; Defense Standardization Program Office, 2012) or aircraft (DO-178C; Radio Technical Commission for Aeronautics, 2012) systems. The paper provides, in table format and related recommendations, examples of two system safety analyses, the FHA and Subsystem Hazard Analysis (SSHA).

A complete list of SSHA LOR task descriptions that arose from our analysis will be available through other Naval Ordnance Safety and Security Activity (NOSSA) documented sources. This research, funded by NOSSA, will provide recommendations to be considered by NOSSA. These recommendations are to help better understand the robustness of the model being developed, especially if that model resides within an SSF. This paper makes recommendations for FHA and SSHA templates and related considerations to facilitate system safety analyses on AI/ML functions with a focus on ways to decrease the autonomy within system operations and increase their correlated SCCs. The questions and discussions devised from this research aim to form guidance and offer best practices to address AI/ML system safety issues.

Use Case to Investigate

When considering an operational use case to implement within our sandbox development environment, our first step was to create a stakeholder's analysis table, as shown in Table 1.

Table 1. Stakeholder's Analysis Table

#	Name/Org	Type	Want/Need	Concern/Loss	Notes
1	Safety Engineer/NAWCWD DS11000	Analyst	Suite of defined LOR tasks and OQE	Guilt/Liability from loss of life	Knows that AI system is Safety Significant but no LOR tool set available
2	Safety Engineer/Contractor (Weapon System Supplier)	Analyst	Suite of defined LOR tasks and OQE	Guilt/Liability from loss of life	Knows that AI system is Safety Significant but no LOR tool set available
3	Warfighter	User	Assurance of weapon system safety	Guilt/Liability from loss of life	Assumes that AI system is safe; unaware of lack of safety rigor
4	WSESRB Member	Analyst	Suite of defined LOR tasks and OQE	Guilt/Liability from loss of life	Knows that AI system is Safety Significant but no LOR tool set available
5	Program Manager	Sponsor	Assurance of weapon system safety	Guilt/Liability from loss of life	Pressured to meet military requirement; accepts safety risk
6	Civilian or Military Victim of Mishap	Neutral Observer	Safety in Battle Space as Non-Target	Personal Death or Injury	Unaware of Latent Safety Hazard
7	American Public	Neutral Observer	Assurance that weapon systems will not kill or injur friendlies or non-combatants	Anger/Disapproval	"How could this tragedy happen?" "Who is responsible?" "Why was a dangerous weapon system deployed by the US
8	NOSSA, PM	Sponsor, Developers	What processes and policy associated with the various phases of the acquisition cycle will be needed to support system safety for AI/ML software?	NOSSA: Unsafe deployed system, PM: Added cost to retrofit safer	
9	NOSSA	Sponsor	What tools, guidance and documentation would need to be created to support the processes and policy per each group's needs? Groups: Developers need from system safety, System safety practitioners from system safety and Oversight folks from system safety.	NOSSA: Unsafe deployed system	
10	NOSSA	Sponsor	Along with the processes, what analytics need investigation for each user group?	NOSSA: Unsafe deployed system	
11	NOSSA	Sponsor	How would various AI/ML software designs affect the analytical approach?	NOSSA: Unsafe deployed system	
12	NOSSA	Sponsor	What kind of OQE is required per a given AI/ML technique and implementation structure to support a program moving forward?	NOSSA: Unsafe deployed system	
13	NOSSA	Sponsor	Will data and analytics be considered as separate pieces to inspect?	NOSSA: Unsafe deployed system	
14	NOSSA	Sponsor	During a WSESRB or Technical Review Panel review that involves AI/ML, how would systems, data and numbers be presented to allow for proper investigation and analysis to ensure contextual accuracy based on group technical background?	NOSSA: Unsafe deployed system	
15	NOSSA	Sponsor	What are the factors and limitations associated with confidence of numbers presented regarding AI/ML performance?	NOSSA: Unsafe deployed system	
16	NOSSA	Sponsor	AI/ML performance is always associated within the context of the training data?	NOSSA: Unsafe deployed system	
17	NOSSA	Sponsor	What does it mean to perform architecture, design, or code analysis (see MIL-STD-882E Table V) with an AI/ML system, especially when, for example, even the developer has limited understanding on how the neural network works?	NOSSA: Unsafe deployed system	
18	NOSSA	Sponsor	How will confidence be assured for each user group in terms of how the software will perform as specified to AI performance requirements (see MIL-STD-882E paragraph 4.4.1.b)?	NOSSA: Unsafe deployed system	
19	NOSSA	Sponsor	What would be the type of contractual language associated with AI/ML integration/deployment?	NOSSA: Unsafe deployed system	
20	NOSSA	Sponsor	Should it include the complete system because of potential reduction in overall system maturity?	NOSSA: Unsafe deployed system	
21	NOSSA	Sponsor	Will AI/ML algorithms exponentially increase the complexity of the system under review affecting hardware issues involved with processing, bandwidth and storage? If not considered, will performance degrade, causing system safety concerns? How will this be analyzed? What are the limitations associated with confidence of numbers presented regarding AI/ML performance? Note: AI/ML performance is always associated within the context of the training data.	NOSSA: Unsafe deployed system	
22	NOSSA	Sponsor	What format will allow technical and non- AI/ML technical stakeholders to support discussion, understanding and eventual application for their particular AI/ML situation? This sets the requirement for how processes and policy should be technically written and displayed while still supporting the necessary detail. It is anticipated that each group will have a different set of requirements for communicating and displaying technical detail related to guidance. What will be the training requirements for each group?different set of requirements for communicating and displaying technical detail related to guidance. What will be the training requirements for each group?	NOSSA: Unsafe deployed system	
23	NOSSA	Sponsor	1. how do we build confidence in the AI black box?	NOSSA: Unsafe deployed system	
24	NOSSA	Sponsor	2. How do we build rigor into, or is it necessary to build rigor into, the training code for the AI?	NOSSA: Unsafe deployed system	
25	NOSSA	Sponsor	Is this the appropriate AI technique to use and is there an non-AI technique that could be used?	NOSSA: Unsafe deployed system	

From a review of Table 1, it became obvious that current system safety analysis methodologies (MIL-STD-882E; Defense Standardization Program Office, 2012) were inadequate to address the unique system safety needs of AI/ML functions. New methodologies would be required to ensure comprehensive system safety analysis of AI/ML functions. In order to conduct research to develop new methodologies, the effort focused on developing a fictional system that implemented various AI/ML functions, allowing the system safety team to investigate gaps in current methodologies when analyzing these specially developed functions. The fictional system would have to replicate parts of the acquisition cycle in detail. To support realism in our fabricated system, various ongoing development efforts within various project and programs in early to late research phases were modified and then combined into an ML hybrid mission planner and a multi-ML algorithm robot technology. Again, it should be emphasized that although the program is fictitious, technology architecture, design, code, and test were based on existing research in the field of AI/ML currently being performed by various naval commands.

An Operational Use Case was identified that had the potential to provide answers described in Table 1's stakeholder analysis. Not only did it need to support answers to the questions posed in Table 1, but the operational used case also needed to be constructed from realistic aspects of AI/ML technology. An operational view of the Use Case is shown in Figure 2.

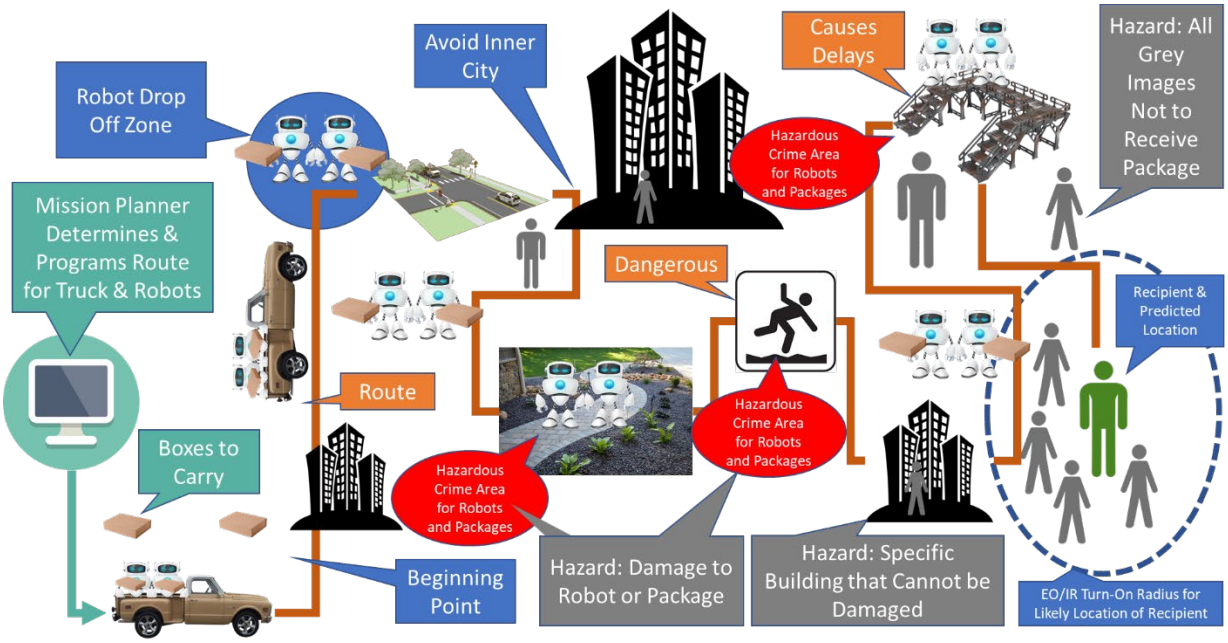


Figure 2. Operational Use Case of Two Robots Delivering Packages

Figure 2 is based on the following considerations involving the Operational Scenario, the Operational/Deployed Environment, and Key SSFs:

The Operational Scenario

- The design consists of the following subsystems: (1) Two Robots, which are identical in performance, (2) Two Pickup Trucks, which are identical in performance, and (3) a Mission Planner.
- The two robots are carried a partial way to their destination on the two pickup trucks. After the pickup trucks arrive at their destination, each robot will be unloaded from their respective pickup trucks. From the unload point, the robots will walk synchronously to arrive and deliver their packages to the single intended recipient at the same time. The two robots are able to walk long distances using GPS navigation, and as the robots gets closer to the subject receiving the package, a Convolutional Neural Network (CNN) image recognition algorithm using color spectrum and infrared images from an EO/IR sensor on the robot will take over navigation. The special GPS navigation is pre-loaded with waypoints produced by an AI/ML trained mission-planning tool.

The Operational/Deployed Environment

- It could be a rainy day when the robots are deployed. Weather conditions, houses, and buildings all result in background clutter and obstacles for the navigation system. There are cars and pickup trucks on the road, including other robots and people walking, complex highway systems, and city-like sidewalks and walkways that need to be navigated by the two robots. Many other people, some looking very similar in side profile to the recipient, in this scenario are part of the environment. It is important that people should not receive this package by mistake, as the packages are hazardous and very valuable. Delivering a

package to the wrong recipient will be a Catastrophic mishap. Thus, it is important to system safety that sufficient mitigations are incorporated into the system to minimize the risk of an incorrect person receiving the package by mistake from either robot.

Key Safety Significant Functions

- Navigation. The navigation system of the robots uses a special GPS function following waypoints and then switches to an AI-based seeker function at a certain range from the recipient. The AI-based seeker function uses polar coordinates instead of waypoints to navigate. The AI/ML navigation system must avoid obstacles, as mentioned previously, during navigation. Some obstacles might include other people attempting to steal the package carried by a robot. The AI-based seeker does not take over navigation until a separate switching function determines when the robot is at a certain distance from the recipient. Once this switch is activated, robot navigation is turned over to the CNN function for final navigation to the recipient. The non-AI navigation is responsible for avoiding obstacles in route until the CNN takes over navigation. Again, once within a certain range of the recipient, the seeker function is switched on to take over the complete navigation of the system. The seeker function consists of a CNN, designed to recognize side profiles of the recipient, and avoid obstacles. The CNN is trained using synthetic images of side profiles within a synthetic clutter environment. The CNN is trained to navigate in such a way as to avoid people attempting to steal its package. While in a traditional system, package theft would be considered a security issue and not a safety issue, the team decided to identify this as a system safety concern to allow investigation of the CNN function.
- Sensor Data. Each robot is receiving non-curated data from a 3-D sensor. The sensor streams a color scaled set of images that contain complex backgrounds at a certain sampling rate for database storage and CNN processing. Images are stored in a separate database for each robot with no data sharing between robots.
- Image Recognition**.** There is a large amount of synthetic data available for training and some actual images of recipients' side profiles. Unfortunately, the added clutter to the image is also synthetic (i.e., building and house backgrounds, day and night lighting, rain, etc.; see previous Operational/Deployed Environment section.) The developer is also considering a transfer learning approach to add another classification layer to the CNN to increase the probability of successful recognition.
- Timing Synchronization. Timing synchronization is implemented using reinforced learning, with real time updates to the Mechanics Reinforced Learning (RL) Dynamics Manager neural network that affects the physics of the robots in terms of direction and speed. Both robots must deliver their packages at the same time, but they can take different routes to avoid environmental conditions. Once a robot delivers a package, the recipient will not wait for the second package. Therefore, it is important that both packages be delivered at the same time to avoid the recipient leaving and the possibility of the second robot delivering its package to someone who should not receive the package. Again, this is a significant system safety concern because delivery of a package to an incorrect recipient is considered a catastrophic mishap.

Sandbox Development Environment Approach

Within the sandbox development environment, a variety of AI/ML algorithms supporting a mission planner and autonomous vehicle selection and navigation were developed—again inspired by existing AI/ML projects and programs. Formal Department of Defense Architecture Framework (DoDAF) (Department of Defense Chief Information Officer [DoD CIO], n.d.; Dam, 2006) and Unified Modeling Language (UML; Booch, 2017) diagrams were created to support a

System of Systems (SoS) design, including interface messages, SQL commands and Application Programming Interfaces (API). Using this sandbox approach allowed an initial top-down safety analysis, starting with system decomposition and traceability from an Operational View (OV), through a Systems View (SV), and then finally down to the algorithm's code level supporting the specific AI/ML being implemented. This process was used to provide a broad scope representation of a potential DoD program implementing AI/ML and to "realistically" investigate conduct of the FHA and SSHA methodologies on a variety of AI/ML functions.

Figure 3 represents the subsystems associated with the mission planner and robot.

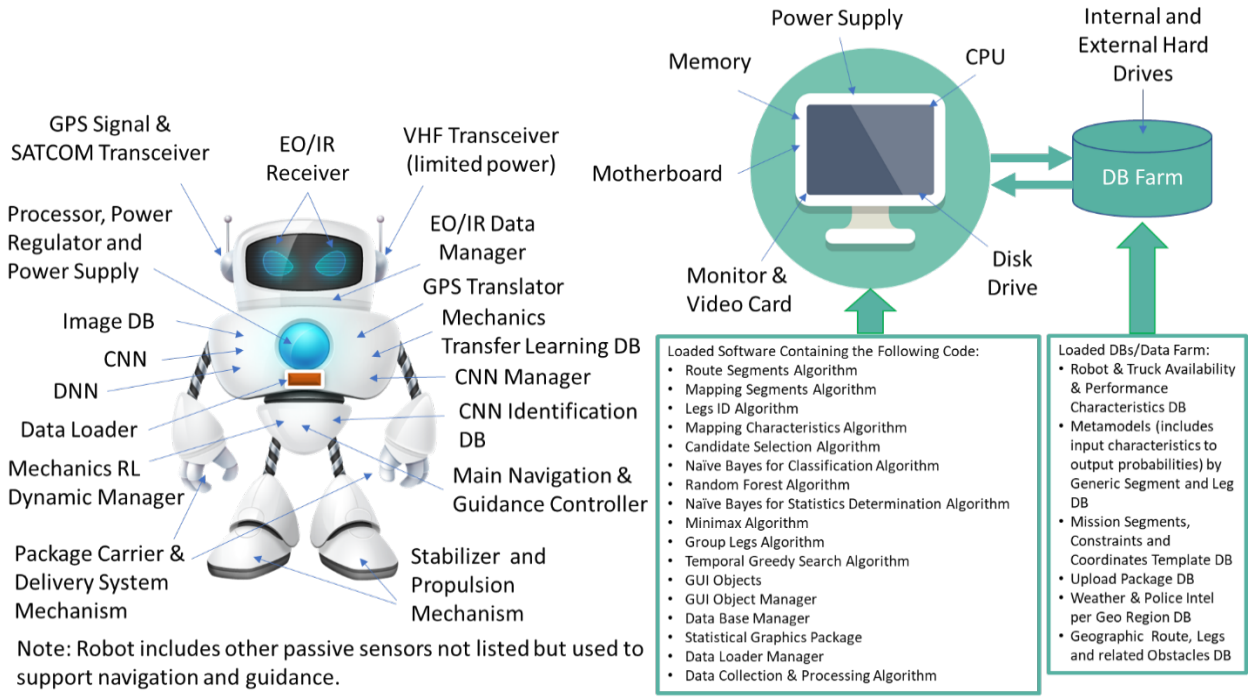


Figure 3. Robot and Mission Planner Subsystems

The goal of the sandbox development environment was to implement a variety of AI/ML technologies that worked together as an SoS, offering a variety of ML approaches to investigate (Hastie et al., 2017). The mission planner provides analysis of the following AI/ML technologies:

- The Naïve Bayes and Logistic Regression algorithms receive sensor and human input data in order to select the appropriate type of meta-model from a repository.
- The Random Forest algorithm, by creating a "Similarity Table" between branches of trees (and its counterpart "Distance Table" for other ML algorithms using distance analysis), allows for data estimation for missing data within the meta-model tables that was not originally identified in the Design of Experiment (DOE) simulation requirements. Therefore, the Random Forest can account for the challenges when the Modeling and Simulation (M&S) is missing inputs needed for a meta-model. Predicting the deployed operational future is difficult for a variety of reasons. Random Forest clustering allows estimation in situations with both limited inputs and an unknown statistical output. This allowed for greater flexibility in the mission planner's ability to adapt in non-ideal operational environments.
- The minimax, per meta-model selected, looks at the factors needing to be addressed in completing the mission, including tactical capability, external challenges and delivery

issues. Based on sensor data, it selects “worst” case scenario and then finds the “best” case autonomous performance combined with tactical sequence needed to successfully complete the mission. “Worst” case and “best” case are represented by statistical structures within the route leg’s meta-model counterpart.

- The meta-model tables are processed through the Random Forest approach and a minimax algorithm to determine the statistics for each leg in the route. Then a non-linear optimization program determines the optimal selection of robots and routes.

The robots’ autonomous platform provides analysis of the following AI/ML technologies:

- Deep Neural Network (DNN) - supports the mechanical motion of the robot given various states. The states are determined based on real time input conditions of the robot movement. Input conditions include traveling surface and conditions. Based on the attributes received, the DNN would use a control feedback mechanism to adjust its walking mechanics. For example, if the robot, through the CNN, recognizes that it is about to approach the package recipient, it would slow down, unsecure the package, and extend its arms to show the package. While traveling, the package would be secure. Upon understanding its current travel state, the robot would use DNN to determine which mechanical state to implement. It should be noted that the sandbox will also be investigating how Deep RL might apply.
- Convolutional Neural Network (CNN) - supports the recognition of the recipient. The input would be based on facial recognition. The result of this analysis would be input to the DNN in terms of its mechanical functions, as discussed previously.

It should be noted that the sandbox is still in development. Design analysis of all AI/ML functions described previously has been completed. Implementation is ongoing. This paper’s findings are based on sandbox development environment investigations, as well as from previous game theory, DNN and CNN research with other projects.

AI Type Definition

There are many opinions surrounding the definition of AI. For this research, we defined the term “AI Type.” We defined an AI Type to be identified by objective measurements. Therefore, if a function is determined to be an AI Type based on its score from the following objective measurements, it requires special FHA and SSHA investigation. The definition and scoring are as follows:

AI Type (Working Definition): For system safety concerns, an AI Type of function means that an algorithm will be developed:

(1) using data approximations to build its algorithm (e.g., from simulations and synthetic data vs. an equation that accurately represents real world physics) and/or

(2) when data samples used to build its algorithm are a subset of the actual population size (e.g., training data samples from population to support machine learning, training data samples requiring clutter backgrounds).

Scoring: For all functions that are candidates for being implemented using an AI/ML algorithm (examples in table), then each function must be graded using criteria (1) or (2), with corresponding points awarded. A final score of 1 or greater indicates that the function is an AI Type.

Table 2 represents examples of scoring based on various AI Types. It is not intended to be a complete list. The goal here is not to provide the system safety practitioner with a complete list, but to aid the practitioner based on a practical scoring approach.

Table 2. Example of Scoring Based on AI Type Definition

AI Type Examples of Specific Algorithms	Algorithm built based on using data approximations	Algorithm built based on using data samples from larger population	Final Score
CNN	x (if synthetic data used for CNN)	x (training data samples)	0 to 2
DNN + SL	x (if synthetic data used for DNN)	x (training data samples)	0 to 2
DNN + RL	x (if synthetic data used for DNN)	x (training data samples)	0 to 2
RNN (LSTM)	x (if synthetic data used for RNN)	x (training data samples)	0 to 2
RNN (Simple)	x (if synthetic data used for RNN)	x (training data samples)	0 to 2
Naïve Bayes	x (if modeling and sim data used to produce statistics for Naïve Bayes)	x (if RL used during opponent interaction to train algorithm)	0 to 2

The AI Type definition allows for a productive, focused discussion between the system safety practitioner and the function developer. Questions that might initiate the conversation would include:

- What parts of the system need special rigor consideration (as compared to traditional algorithm code development)?
- Does the SSF identified qualify as an AI/ML function?

It is especially important to investigate if the function is an AI Type when an algorithm is identified as a safety-critical function using an FHA approach. Again, the hypothesis is that it is an AI Type function if

- (Consideration 1) it uses data approximations to build/train its algorithm (e.g., data approximations can come from simulations and synthetic data vs. an equation that accurately represents real world physics), and/or
- (Consideration 2) data samples are used to build/design its algorithm and these data sample are a subset of the actual population size (e.g., training data samples from population to support machine learning, training data samples requiring clutter backgrounds).

One way to think about consideration (1) is to ask, “Could another developer create a different set of statistics under the same conditions?” If no, then maybe this algorithm is not an AI Type. If yes, then it meets the condition. As an example, if a statistical model of the function was developed, how accurate were the approximations used in creating the function. In other words, how close do these approximations fit the real world physics regarding operational deployment? If the function is based on simulation results, then the concern is the “garbage in, garbage out” issue—poor real world representative synthetic data will result in an inferior model. The goal is to have good quality and comprehensive training data that would result in a robust model.

One way to think about consideration (2) is to ask, “What is the actual population size of the training set?” If the training set is equal to the actual population size, then it is not an AI Type. Consider the most basic ML algorithm, a regression line. If all the points that will ever occur for this function are on the scatter plot used to approximate the curve, why use a regression line? If all the ML algorithm inputs and outputs are known, why use ML and not traditional code? Again, if traditional code can address the needs of the function, then that would be the goal.

Notice that both considerations are related. It is like looking at two sides of the same coin: how ML algorithms are developed and why they need to be avoided in critical functions.

Figure 4 describes the need to separate out AI Type designated functions from traditionally developed software coded functions.

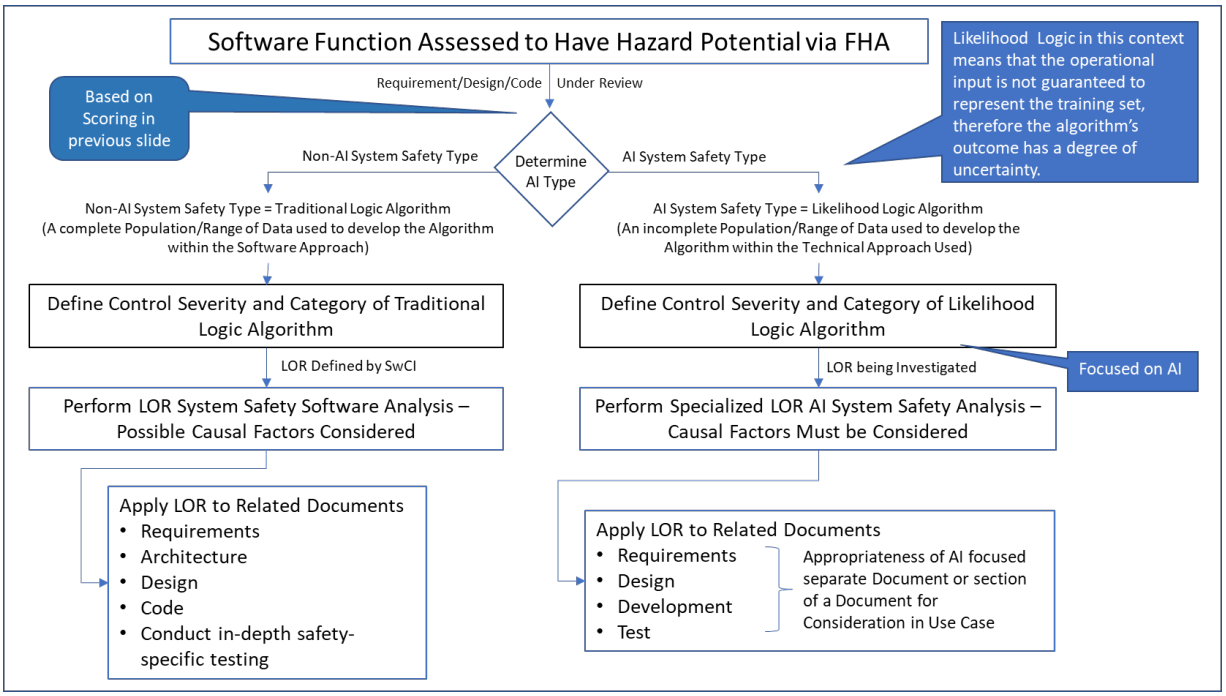


Figure 4. Flow to Assess AI Type Using Special FHA and SSHA Rigor

Six Recommendations to Assess AI Type Functions

This paper describes six recommendations. The first three are associated with the FHA assessment, and the last three involve the SSHA analysis. The FHA recommendations are provided in the form of a complete list of columns to use during the development of the FHA. It is noted that the current FHA approach works well in identifying the safety significance of a function. As shown in Table 3, three columns have been added in support of the three recommendations focused on reducing autonomy. The last three recommendations are in support of doing an SSHA. One recommendation is in structuring the table, with an added column regarding the focus of the analysis. The final two recommendations offer a list of questions listed as line items within the table focused on (1) API/MSG/SQL interface corruption to the ML algorithm and (2) modality of how well the data training the algorithm represents deployed conditions. The fifth recommendation addresses the interface to the algorithm and asks questions involved with how data corruption might need to be addressed within the training data composition of the ML algorithm. The sixth recommendation is in regards to the modality of how the training data is gathered, organized and managed, known as curation. If the training data isn't curated properly, the ML algorithm may not function properly. In either case, interface corruption or poor curation, a safety critical ML function could have a greater likelihood of becoming a hazard.

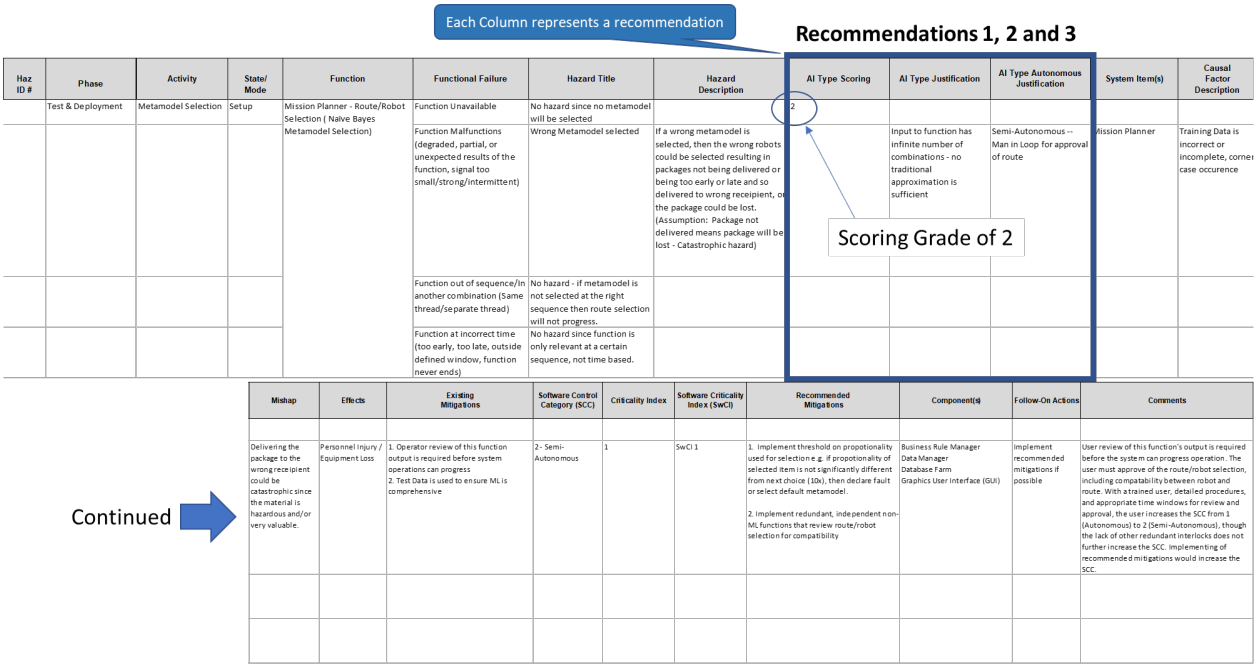
Functional Hazard Analysis

Prior to addressing the first three recommendations, the initial steps of the functional hazard analysis (FHA) need to have been completed to identify the safety significance of the functions (JS-SSA Rev. A, 2017). It should be noted that the system safety analysis found no gaps in the FHA process in identifying the safety significance of a function when dealing with an AI/ML function. This is because the identification of the Severity, SCC, and resulting SwCI does

not change when dealing with an AI/ML function. At this stage, it does not matter whether the function has AI/ML capability in it or not.

An example of an FHA regarding our Use Case example for mission planning is described in Table 3.

Table 3. FHA Example for Mission Planner From Sandbox



Haz ID #	Phase	Activity	State/ Mode	Function	Functional Failure	Hazard Title	Hazard Description	AI Type Scoring	AI Type Justification	AI Type Autonomous Justification	System Item(s)	Causal Factor Description
	Test & Deployment	Metamodel Selection	Setup	Mission Planner - Route/Robot Selection (Naïve Bayes Metamodel Selection)	Function Unavailable	No hazard since no metamodel will be selected		2				
					Function Malfunctions (degraded, partial, or unexpected results of the function, signal too small/strong/intermittent)	Wrong Metamodel selected	If a wrong metamodel is selected, then the wrong robots could be selected resulting in packages not being delivered or being too early or late and so delivered to wrong receipient, or the package could be lost. (Assumption: Package not delivered means package will be lost - Catastrophic hazard)		Input to function has infinite number of combinations - no traditional approximation is sufficient	Semi-Autonomous -- Man in Loop for approval of route	Mission Planner	Training Data is incorrect or incomplete, corner case occurence
					Function out of sequence/In another combination (Same thread/separate thread)	No hazard - if metamodel is not selected at the right sequence then route selection will not progress.						
					Function at incorrect time (too early, too late, outside defined window, function never ends)	No hazard since function is only relevant at a certain sequence, not time based.						

Mishap	Effects	Existing Mitigations	Software Control Category (SCC)	Criticality Index	Software Criticality Index (SwCI)	Recommended Mitigations	Component(s)	Follow-On Actions	Comments
Delivering the package to the wrong/receipient could be catastrophic since the material is hazardous and/or very valuable.	Personnel Injury / Equipment Loss	1. Operator review of this function output is required before system operations can progress 2. Test Data is used to ensure ML is comprehensive	2- Semi-Autonomous	1	SwCI 1	1. Implement threshold on propotionality used for selection e.g. if propotionality of selected item is not significantly different from next choice (10x), then declare fault or select default metamodel. 2. Implement redundant, independent non-ML functions that review route/robot selection for compatibility	Business Rule Manager Data Manager Database Farm Graphics User Interface (GUI)	Implement recommended mitigations if possible	User review of this function's output is required before the system can progress operation. The user must approve of the route/robot selection, including compatability between robot and route. With a trained user, detailed procedures, and appropriate time windows for review and approval, the user increases the SCC from 1 (Autonomous) to 2 (Semi-Autonomous), though the lack of other redundant interlocks does not further increase the SCC. Implementing of recommended mitigations would increase the SCC.

Recommendation 1: Once an FHA identifies the SSFs, each function needs to be assessed using the AI Type definition following these three steps:

(1) For each identified SSF that potentially has AI/ML within, identify the algorithm that will be used to support that function, and then document the grade each function receives in terms of AI Type definition (see previous section on AI Type scoring and definitions).

(2) If the function scores either 1 or 2 (i.e., it is an AI Type), describe what specific AI/ML algorithm is going to be trained/implemented.

(3) If the specific algorithm is using DNN structures (i.e., three or more layers), identify if there is enough training data to support this approach and, if not, were older ML algorithms considered, like Logistic Regression, kNNs, etc.

If function qualifies as an AI Type, follow these two recommendations:

Recommendation 2: Verify that an AI/ML function is needed by asking the following questions:

a) For AI Type definition 1: Can the algorithm be traditionally built using data approximations? Why or why not?

b) For AI Type definition 2: Can the algorithm be broken into subpopulations to allow development of traditional code? Why or why not?

Recommendation 3: Justify that an AI/ML function needs to be Autonomous by documenting the following:

a) Document how the design can or cannot include a human in the loop or traditional hardware/software technology to provide checks and balances.

b) If it cannot provide checks and balances, provide documentation as to operational limitations by:

1. Describing weaknesses of each AI/ML technique (e.g., expected success rate of the function). For example, if AI/ML is built on data approximations (using AI Type definition), how much bias will the data approximations add to the functional outcome? Or if AI/ML is built on data samples (using AI Type definition), how representative are the samples to the population?
2. Determining how the training data is being generated (e.g., truth, synthetic, combination). Are these sources valid? Why?
3. Where is the training data coming from? Is it enough? (Remember the more sophisticated the AI/ML software, the more likely that it needs larger amounts of training data.)
4. Will an outside independent source review the training, validation and test data created? Why or why not?
5. Will an outside independent source validate the success rate of the AI/ML function as compared to other AI/ML functions used in industry? Why or why not?

SSHA

Table 4 and Table 5 show analysis of one of the AI/ML functions in the sandbox, a 17-attribute, five-class Naïve Bayes algorithm for meta-model selection that implements statistically independent instances representing missing and sparse data operational issues. In our sandbox, we used 15,000 samples of training data to support classification training of Logistic Regression and Random Forest algorithms. Our analysis is independent of the categorization algorithm selected but focuses on how to assess it in terms of identifying hazards at the subsystem level. Naïve Bayes is used in the example to remove focus on the algorithm complexity and place it on the recommendations being offered.

Recommendation 4: Table 4 identifies the hazard description and, again, provides a similar table approach to non-AI functions under investigation.

Table 4. SSHA for Meta-Model Selection Algorithm Within Mission Planner From Sandbox

Haz ID#	Phase	State/ Mode	System	Subsystem	Component	Element	Hazard Title	Hazard Description	Causal Factor Description	Mishap	Effects	Existing Mitigations
Identifier used to reference specific hazard.	*The life-cycle phase for which the risk and risk assessment apply. Multiple phases can be specified if the risk & mitigations are equivalent.*	*The State and/or Mode of the system for the hazard of concern.*	*The composite at any level of complexity of interworking parts (personnel, procedures, equipment, hardware, software, et al) used together to perform a task or accomplish a mission.*	*A functional or physical portion of a system designed, used or integrated to accomplish one aspect of the system task or mission.*	*A functional or physical portion of a subsystem designed, used or integrated to accomplish one aspect of the subsystem task or objective.*	*A functional or physical portion of a component designed, used or integrated to accomplish one aspect of the component .*	*Short title of the hazard*	*The detailed description of the conditions under which hazardous energy may be released in an uncontrolled or inadvertent way.*	*The detailed description of the failures, conditions, or events that contribute either directly or indirectly to the existence of the hazard.*	*The event or series of events where hazardous energy release could negatively effect equipment, personnel or environment; accident*	*The results of the mishap to include injury or death, damage to equipment and property, or damage to the environment.*	*Controls that are already planned existing to mitigate the risk.*
SSHA-001	Test & Deployment		Mission Planner				AI Function for metamodel selection (Naïve Bayes) failure	If a wrong metamodel is selected, then the wrong robots could be selected resulting in packages not being delivered or being too early or late and so delivered to wrong receipient , or the package could be lost. (Assumption: Package not delivered means package will be lost - Catastrophic hazard)	Inadequate quality or quantitiy of training data.	Delivering the package to the wrong receipient could be catastrophic since the material is hazardous and/or very valuable.	Personnel Injury / Equipment Loss	
									Incorrect algorithm selection			
									Improper curation of data			Multiple sources (primary, secondary and tertiary sources) accommodate sources that fail/missing.
									Too much or too little data (Underfitting and Overfitting of model)			

Recommendation 5: For system safety practitioners, Table 5 might also look familiar. The recommendation is to add a single column labeled “Focus” that categorizes the LOR list of descriptions that might be unique to ML algorithm development. Notice that the list of LOR Descriptions is based on the “Focus” described. It is a simple recommendation, but from our research within the sandbox, it helps organize the range of issues that might occur. For every AI Type identified in the FHA, it is recommended that an interface analysis be considered, as described in Table 5, using the LOR Description column. Each row in this LOR Description column provides API/MSG/SQL interface questions that might affect the algorithm’s performance during deployment, as will be explained next.

Table 5. SSHA LOR Table Example Based on Data Flow Analysis of Meta-Model Selection Within the Mission Planner

Level or Rigor (LOR) Activity	Phase	Focus	LOR Description	Primary Responsibility	Support Responsibility	Baseline	Software Criticality Index (SwCI)				Representative Artifacts Produced
							4	3	2	1	
ALG6: Data Flow Analysis for the Mission Planner	Algorithm Design, Algorithm Code and Test and Evaluation	API/MSG/SQL Interface	Would the corruption of API/MSG/SQL/Other affect data variations requiring additional training of the Target Algorithm?	AI/ML Algorithm Developer	Data Analytics Engineer						Data Analytics Report
			If so, will quality (composition/complexity/structure) of Training Data significantly increase? Explain specific to the API/MSG/SQL/Other.	AI/ML Algorithm Developer	Data Analytics Engineer						Data Analytics Report
			Will these variations be part of the analysis for selecting the "best" algorithm? Explain.	AI/ML Algorithm Developer	Data Analytics Engineer						Data Analytics Report
			Because of this issue, will quantity (more instances) of Training Data significantly increase? Explain specific to the API/MSG/SQL/Other.	AI/ML Algorithm Developer	Data Analytics Engineer			R	R	R	Data Analytics Report
			Will creating/finding enough training data replicating the corruption be an issue? Explain.	AI/ML Algorithm Developer	Data Analytics Engineer						Data Analytics Report
			Are you confident that any additional data created/found will adequately represent the effects associated with replicating the corruption? Explain.	AI/ML Algorithm Developer	Data Analytics Engineer						Data Analytics Report
		ML Modality: During Deployment & Curation Congruency	Based on Modality Table: Describe Data Source Precedent for Improving Success Rate (ranking of primary, secondary tertiary... n attributes) -- by addressing related question in the table.	AI/ML Algorithm Developer	Data Analytics Engineer						Training Data Curation Report
			Based on Modality Table: Describe how missing and sparse data issues are modeled -- by addressing related question in the table.	AI/ML Algorithm Developer	Data Analytics Engineer			R	R	R	Training Data Curation Report
			Based on Modality Table: Describe how the quality of Training Data Characterized -- by addressing related question in the table.	AI/ML Algorithm Developer	Data Analytics Engineer						Training Data Curation Report
			Based on Modality Table: Describe how the quantity of Training Data Characterized -- by addressing related question in the table.	AI/ML Algorithm Developer	Data Analytics Engineer						Training Data Curation Report

- PR: Prerequisite Requirement – Required regardless of LOR or required in order to assess and determine LOR
- AD: As directed by Customer/Contract
- R: Required for assigned LOR
- IV&V: Independent Verification and Validation
- N/A: Not Applicable for this program or LOR

To understand whether the ML Algorithm was trained properly to handle issues based on interface corruption, the following six questions (in sequential rows) are recommended based on our sandbox analysis:

1. Would the corruption of the API/MSG/SQL/Other affect data variations requiring additional training of the AI/ML Algorithm? This is a yes or no answer.
2. If yes, will quality (composition/complexity/structure) of Training Data significantly increase? Will it affect the ML Training Modality? Explain this specific to the API/MSG/SQL/Other. Corruption might result in a need to add secondary or tertiary sources. It might also affect how data is collected from various sources, potentially changing the ML Training Modality.
3. Will these variations be part of the analysis for selecting the "best" algorithm? Explain. ROC sweet spot analysis might be used with hyper parameter changes based on the type of variation.
4. Because of this issue, will quantity (more instances) of Training Data significantly increase? Explain this specific to the API/MSG/SQL/Other. This could result in a need to have more of a certain type of instance to train on based on mixes of primary, secondary, or tertiary attribute requirements.
5. Will creating/finding enough training data replicating API/MSG/SQL/Other corruption be an issue? Explain. If it is synthetic, is may not be an issue, depending on the model. If it

comes from “live” data, then would there be more training data associated with the effects of the corruption?

6. Is there confidence that any additional data created/found will adequately represent the effects associated with replicating the corruption? Explain. This is an important statement related to the quality (composition/complexity/structure) of the Training Data.

Recommendation 6: Another series of rows has been added to Table 5 based on modality associated with the training data. Table 6 provides additional questions for the algorithm developer and data analytics engineer to address based on modality regarding how the ML is trained.

Table 6. Investigation Questions Based on Modality

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

A brief summary of modality types that support training data composition and size are described next:

- ML Training Data Modality 1: This modality supports training data sets that are based on an operational environment from multiple data sources, where each source contains one or more attributes. The various sources of separate data attributes are either found from live events or synthetic simulations created to match the deployed operational scenario. Therefore, the input for the ML algorithm for training needs to replicate the input that will be received during deployment.
- ML Training Data Modality 2: Training data sets that are based on an operational environment from a single data source, where the single data source contains multiple data attributes. The one stream set of aggregated attributes is either found from live events or synthetic simulations created to match the deployed operational scenario. Therefore, the input for the ML algorithm for training needs to replicate the input during deployment.
- ML Training Data Modality 3: Training data sets that are based on an operational environment from a combination of multiple data sources where each source contains one or more attributes form various sources and from a single source containing multiple aggregated data attributes. It is a combination of Modality 1 and 2 that the algorithm uses for categorization or regression.

Conclusions and Final Best Practice Recommendations

Our findings indicate that the FHA and SSHA for AI/ML SSFs need to be addressed differently from traditional functional analysis methods. To address these differences, the AI Type definition and scoring approach was introduced, along with six recommendations regarding the FHA and SSHA. The research describes questions/issues needing to be addressed when conducting safety analysis on AI/ML function types. It includes discussion on how the current FHA process is still valid for AI/ML functions and only requires three additional columns to support added justification that an AI/ML function is required to meet operational goals. The SSHA discussion provides a simple table modification and two examples of LOR Descriptions that need to be addressed when dealing with AI/ML critical functions: (1) interfaces to the algorithm to understand the impact of potential data corruption, and (2) modality issues to ensure robust curation of the data to ensure the algorithm is trained to meet deployment challenges.

Along with the AI Type scoring and six recommendations associated with the analysis of critical functions, there were complementary "Best Practice" questions that arose from our sandbox development environment when developing AI/ML algorithms within a deployed weapons system.

"Best Practice" areas to consider specific to AI/ML critical functions include:

General AI/ML Questions:

- When a critical function is identified, does it meet the AI Type definition criteria: (1) Is the algorithm built based on using data approximations, and/or (2) is the algorithm built based on using data samples from larger populations?
- When doing M&S to create the training data, does the simulation adequately represent Classes for the ML process? If not, how are Classes represented?
- Does each Class have a sufficient number of attributes that can be learned by the algorithm for that Class? Are overfitting and underfitting considered for that Class with regards to the quantity of attributes simulated and does that reflect real world operations?
- How do we know that the M&S creating the training data is aligned with the mission parameters? Was a traceability study performed to ensure adequate coverage? Have statistics been developed to show how many configurations exist and how many were trained using primary, secondary, and tertiary data sources? How are we avoiding overfitting and underfitting based on primary, secondary, and tertiary training data mixes and sets? Is the training data organized in terms of primary, secondary, and tertiary attributes to be able to represent missing and sparse data priorities from related sources?
- How are we ensuring that the algorithm being deployed provides the correct answer when data input issues occur? Is the algorithm success rate determined by primary, secondary, and tertiary attributes?
- Can other control entities (such as a human operator) be inserted into the loop to reduce the SCC?

Operational "Realism" Questions:

- Is the M&S able to create training data that represents reality when sparse and missing data issues occur?
- Does the architecture, design, and code support sparse and missing data management; specifically, does it filter or select less significant attributes to do the calculations?

- Does the data management support filtering to ensure the ML algorithm is provided accurate data input, avoiding “garbage in, garbage out?” Has what constitutes “garbage in, garbage out” been defined?
- How well does the particular ML algorithm support increased complexity, and how does that affect sparse and missing data issues?

Selected AI/ML Algorithm:

Note: Individual types of AI/ML require specific questions that address their method and application. Naïve Bayes is used as a simple example, but Logistic Regression, Random Forest, DNNs, or other categorization algorithms could have used the training data produced within our sandbox environment.

Some questions to guide the examination of Naïve Bayes are:

- How do you trust the behavior in the real world for this Naïve Bayes function? Success Rate? Quality of Training Data? How did it compare with other categorization algorithms?
- Was Naïve Bayes the correct selection for this function vs. other algorithms? The choice should be based on what gives you the best operational performance and understanding of operational limits (Potential OQE: k-fold cross validation comparisons, etc.). How reliable will the answers be in the real world?
- How do you assess the operational limits of this Naïve Bayes (or alternate algorithm) categorization function?
- Did the training set model enough noise/clutter (in this case, less significant attributes determined by SMEs for a particular meta-model class) for each class that allows for the function to work properly when deployed? Are there sparse data and/or missing data issues? How is the bias of the training set and variance of the test determined?
- How would you ensure simulation configurations (i.e., the training data) are adequately covering the real world experiences? Consider optimizing bias (how well it fits the training set) and variance (how well it predicts using the test set)—overfitting/under-fitting.
- How many types of simulations and how much training data is really enough?
- Are the attributes used for the assessment really independent?
- Is the size of the alpha correct? Is this hyper parameter optimally used?
- Is MAP or Maximum Likelihood better for this calculation?

Developing defined lists of questions/issues, as described in this paper, allows system safety professionals to identify how to increase the inherent safe operation of safety significant AI/ML functions. By following the guidance provided, the system safety practitioner can drive important discussions on the development of the AI/ML function and thereby potentially influence design of the overall system to decrease the mishap risk associated with these specially developed functions.

References

Booch, G. (2017). *Unified modeling language user guide* (2nd ed.). Addison-Wesley Professional.

Brose, C. (2020). *The kill chain: Defending America in the future of hi-tech warfare*. Hachette Books.

Dam, S. (2006). *DoD architecture framework: A guide to applying system engineering to develop integrated, executable architecture*. SPEC.

Defense Standardization Program Office. (2012). *System safety* (MIL-STD 882E). Pentagon.

DoD Chief Information Officer. (n.d.). *The DoDAF architecture framework version 2.02.* https://dodcio.defense.gov/library/dod-architecture-framework/

Hastie, T., Tibshirani, R., & Friedman, J. (2017). *The elements of statistical learning data mining, inference and prediction* (2nd ed.). Springer.

Joint Services – Software Safety Authorities. (2017). *Software system safety implementation process and tasks supporting MIL-STD-882E* (JS-SSA-IG Rev. A).

Joint Software Systems Safety Engineering Workgroup. (2017). *Software system safety implementing process and tasks supporting MIL-STD-882E* (JS-SSA-IF Rev. A). Pentagon.

National Defense Authorization Act (NDAA) for Fiscal Year 2021, Pub. L. No. 116-283 (2021).

National Institute of Standards and Technology. (2019). *U.S. leadership in AI: A plan for federal engagement in developing technical standards and related tools*. https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf

National Security Commission on Artificial Intelligence. (2019). *Final report: Establishing justified confidence in AI systems* (Ch. 7).

National Security Commission on Artificial Intelligence. (2021). Autonomous weapon systems and risks associated with AI-enabled warfare. In *Draft final report* (pp. 45–53). https://www.nscai.gov/wp-content/uploads/2021/01/NSCAI-Draft-Final-Report-1.19.21.pdf

Naval Sea Systems Command. (2008). *Department of the Navy Weapon Systems Explosives Safety Review Board* (NAVSEAINST 8020.6E).

Radio Technical Commission for Aeronautics. (2012). *Software considerations in airborne systems and equipment certification* (DO-178C). Federal Aviation Administration.

Sodhani, S. (2018). *A summary of concrete problems in AI safety*. https://futureoflife.org/2018/06/26/a-summary-of-concrete-problems-in-ai-safety/

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

www.acquisitionresearch.net