

RATIONALE

In order to provide a means of compliance for the certification of Artificial Intelligence (AI) within safety critical aeronautical systems, the committee must first review existing standards and perform a gap analysis to understand how and why existing standards cannot be reliably used. This document serves as that gap analysis and provides a list of concerns that need to be addressed in order to produce a future means of compliance.

TABLE OF CONTENTS

1.	SCOPE	5
2.	REFERENCES AND DEFINITIONS	5
2.1	References	5
2.2	Definitions	8
3.	CLASSIFICATION OF AI TECHNIQUES	13
3.1	Symbolic AI	14
3.1.1	Logic	14
3.1.2	Numerical Analysis	15
3.1.3	Machine Learning (ML)	16
3.2	Machine Learning System Workflow	18
4.	GAP ANALYSIS FROM EXISTING STANDARDS	19
4.1	Introduction	19
4.1.1	Description	19
4.1.2	Gap Analysis Methodology	20
4.2	A Hypothetical Development Scenario	20
4.2.1	Machine Learning Technique	20
4.2.2	Development Life Cycle Activities Description	20
4.3	Standards Analysis	21
4.3.1	ED-79A/ARP4754A Guidelines for Development of Civil Aircraft and Systems	21
4.3.2	ARP4761 Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment	21
4.3.3	ED-12C/DO-178C Software Considerations in Airborne Systems and Equipment Certification	22
4.3.4	ED-218/DO-331 Model-Based Development and Verification Supplement to ED-12C	23
4.3.5	ED-215/DO-330 Software Tool Qualification Considerations	24
4.3.6	ED-216/DO-333 Formal Methods Supplement to ED-12C and ED-109A	24
4.3.7	ED-217/DO-332 Object-Oriented Technology Supplement to ED-12C and ED-109A	24
4.3.8	ED-80/DO-254 Design Assurance Guidance for Airborne Electronic Hardware	24
4.3.9	ED-109A/DO-278A Software Integrity Assurance Considerations for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems	25
4.3.10	ED-153 Guidelines for ANS Software Safety Assurance	25
4.3.11	Commission Implementing Regulation (EU) 2017/373	26

SAE Executive Standards Committee Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2021 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
http://www.sae.org

SAE WEB ADDRESS:

Full document details: SAE AIR6988

For more information on this standard, visit
<https://www.sae.org/standards/content/AIR6988>

Documents

Document Sharing Hub

4.3.12	ISO 12207 Systems and Software Engineering - Software Life Cycle Processes	27
4.3.13	ISO 26262 Road Vehicles - Functional Safety	27
4.3.14	ISO 21448 Road Vehicles - Safety of the Intended Functionality	27
4.3.15	ED-201 Aeronautical Information System Security (AISS) Framework Guidance	27
4.3.16	ED-202A/DO-326A Airworthiness Security Process Specification	28
4.3.17	ED-203A/DO-356A Airworthiness Security Methods and Considerations	28
4.3.18	ED-204/DO-355 Information Security Guidance for Continuing Airworthiness	28
4.3.19	ED-205 Process Standard for Security Certification and Declaration of ATM ANS Ground Systems	29
4.4	Gap Analysis Summary	29
5.	ML DEVELOPMENT SPECIFIC CONSIDERATIONS AND AREAS OF CONCERNS	29
5.1	Criteria for the Identification of Specific Considerations and Areas of Concerns	29
5.2	Specific Considerations and Areas of Concerns	30
5.2.1	System Definition	30
5.2.2	Data Selection and Validation	30
5.2.3	Model Selection, Training, and Testing	31
5.2.4	Inference Implementation of the ML Model	32
5.2.5	ML Sub-System Verification	32
5.2.6	System Integration and Verification	33
5.2.7	Operation	33
6.	POTENTIAL NEXT STEPS	33
6.1	Scope of Work	33
6.1.1	Airborne and Ground Systems	33
6.1.2	AI Techniques Included	34
6.1.3	Autonomy	34
6.1.4	Cybersecurity	34
6.1.5	Trustworthiness Analysis	35
6.2	Suggested Approach for Certification/Approval of Systems Utilizing AI	36
6.2.1	Phases of the Product Lifecycle	36
6.2.2	Potential AI/ML Development Assurance Activities	40
6.2.3	Potential AI/ML Development Assurance Process	43
7.	USE CASES - AIRCRAFT SYSTEMS	44
8.	AIR TRAFFIC MANAGEMENT (ATM)/GROUND SYSTEMS OPERATIONS (GSO)	51
9.	CONCLUSIONS AND WAY FORWARD	59
10.	NOTES	61
10.1	Revision Indicator	61
Figure 1	AI classification	13
Figure 2	The spectrum of ontology kinds [34]	14
Figure 3	Machine learning bubble chart	17
Figure 4	Example of A SNN [66]	18
Figure 5	Example of A DNN [66]	18
Figure 6	Machine learning system workflow	18
Figure 7	Product lifecycle for ML-based products	37
Figure 8	Development assurance process for certification/approval	43
Figure 9	ML workflow in a system development	60
Figure 10	SAE G-34/EUROCAE WG-114 joint committee structure	61

Table 1	Example use cases for aircraft systems	45
Table 2	Air space traffic management use-cases	52
Table 3	Ground traffic management use cases	55
Table 4	Airworthiness use cases	56
Table 5	Air traffic communications use cases.....	56
Table 6	Remote ID from NAS infrastructure use cases.....	56
Table 7	Mixed man and unmanned operations use case	57
Table 8	Predictive maintenance use cases	58

PREFACE

Anticipating a growing commercial pressure for Artificial Intelligence (AI) solutions within the aerospace industry over the coming few years, there is an urgent call for regulation and the emergence of norms around acceptable usage. In response, two working groups were set up independently on either side of the Atlantic during 2019 to address concerns around assuring products and services that exploit AI technologies. WG-114 was established by EUROCAE in Europe and G-34 by SAE in the United States.

Both Working Groups were created to produce guidance on safe, secure, and successful adoption of AI technologies in Aeronautical Systems, through consensus amongst many experts and practitioners in industry and academia. In bilateral agreement, the groups formed a joint committee in June 2019.

The joint working group will evaluate key applications for AI usage within aeronautical systems, with a scope encompassing ground-based equipment and airborne vehicles, including Unmanned Aircraft Systems (UAS) products. In terms of processes, the full lifecycle will be under consideration, from design and manufacture, to operation and through-life maintenance.

A key deliverable will be documented standards, providing guidance on assuring safe and secure systems utilizing AI, through an agreed acceptable means of compliance with regulatory requirements.

As per the charters of both EUROCAE WG-114 and SAE G-34, the first objective of the joint working group was to develop and publish a technical report, a comprehensive Statement of Concerns (SOC), outlining the scope and purpose of the group's work and considering the concerns before imagining the solutions. This document is the response to that objective and is the outcome of virtual meetings that took place between September 2019 and May 2020.

Before merging, each of the groups was developing a SOC document independently. Both groups were organized into sub-groups, each addressing a section of the SOC. G-34 had Sub-Committees (SCs) and WG-114 had Teams (TMs). Upon merging, six groups were formed, as defined in the following table.

WG-114	G-34	Scope
TM1	SC4	Taxonomy/Classification of AI Techniques
TM2	-	Areas of Concerns
TM3	SC1	Gap Analysis from Existing Standards
TM4	-	Potential Next Steps
-	SC2	Aircraft Systems
-	SC3	Ground Use-Case & ATO

AI has the potential to disrupt the aerospace industry, impacting all areas in which computing and aerospace intersect. AI technologies are becoming progressively more embedded into the digital systems used to design, manufacture, operate, and maintain both aerial vehicles and ground-based systems. Leveraged appropriately, AI-driven solutions could transform the products and services that aerospace companies provide with an accelerated pace of change. Specifically, Machine Learning (ML) technologies have the potential to revolutionize established paradigms of aeronautical system development, including those concerned with safety-critical applications.

AI is a broad subject, still being actively developed from a confluence of many disciplines, including mathematics, computing, cognitive science, software development, data science, control theory, and others. It demands a collaborative approach with experts contributing from multiple domains.

Current industry guidance documents have a strong focus on established development methodologies and solutions for aeronautical systems. These standards are therefore not expected to entirely accommodate the development and assurance of ML-enabled solutions.

SUMMARY OF CONTENTS

Section 3 surveys established AI techniques and their application to various problem domains, placing them in a historical context and suggesting a taxonomy. A development workflow is also outlined.

From the foundational background provided by Section 3, Section 4 assesses current aerospace industry standards and guidelines, discussing their applicability to systems that incorporate ML technologies. Chief among these standards are ED-12C/DO-178C and supplements, ED-79/ARP-4754A, ED-80/DO-254, ED-109A/DO-278A, and ED-153.

In Section 5, specific areas of concern for both developmental and operational activities are identified, using the development workflow outlined in Section 3 as a framework for discussion. The scope is limited to technical considerations. Ethical and societal implications are out of scope.

Section 6 defines the scope of the documents that the joint Working Group will produce. Both ground and airborne applications are considered, at whole-system level, rather than at the level of their constituent hardware and software components. Software operates in the context of a system, and key quality attributes such as safety are properties of the overall system.

Potential applications of AI to aerospace domains are proposed as a set of Use Cases in Sections 7 and 8, for airborne and ground-based systems, respectively.

The conclusion in Section 9 proposes a staged plan for developing a future standard for AI techniques, as well as processes and technologies to be assessed against, in the context of aeronautical systems.

IMPORTANT NOTICE: This document neither defines guidance, nor sets forth any constraints around future guidance.

1. SCOPE

This document reviews current aerospace software, hardware, and system development standards used in the certification/approval process of safety-critical airborne and ground-based systems, and assesses whether these standards are compatible with a typical Artificial Intelligence (AI) and Machine Learning (ML) development approach. The document then outlines what is required to produce a standard that provides the necessary accommodation to support integration of ML-enabled sub-systems into safety-critical airborne and ground-based systems, and details next steps in the production of such a standard.

NOTE: This document and the upcoming standard it is to help inform are concerned only with “offline” learning applications of AI and ML. In offline learning, ML models are trained on historical data within a dedicated learning environment. When the trained models are then implemented into a production system, learning functionality is turned-off. The production system implementing AI may collect data for retraining at a later date, but any retraining or further learning will happen in the separate learning environment, and any resulting changes to the ML models will then need to be re-implemented into production as a new version of the system utilizing AI. This is in contrast to “online” learning, where a system utilizing AI will continue to learn and adapt its operation while in production. Consideration of such systems is not out of scope for SAE-G34, but the committee will not consider online learning until after publication of this document and its related standard.

2. REFERENCES AND DEFINITIONS

2.1 References

- [1] NASA/Honeywell, “DOT-FAA-TC-16-4 Verification of Adaptive Systems,” 2016.
- [2] EASA, “EASA AI Roadmap 1.0,” 2020.
- [3] SAE, “ARP6128 - Unmanned Systems Terminology Based on the ALFUS Framework,” 2019.
- [4] SAE, “J3016 - Taxonomy and Definitions for Terms related to Driving Automation Systems for On-Road Motor Vehicles,” 2018.
- [5] A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow; 2nd ed, O'Reilly Media, 2019.
- [6] ISO/IEC 2382:2005, “Information Technology - Vocabulary,” 2015.

- [7] SAE/EUROCAE, "ARP4754A/ED-79A - Guidelines for Development of Civil Aircraft and Systems," 2010.
- [8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, The MIT Press, 2016.
- [9] RTCA/EUROCAE, "DO-178C/ED-12C Software considerations in airborne systems and equipment certification," 2013.
- [10] High-Level Expert Group on AI (AI HLEG), "Ethics Guidelines for Trustworthy AI," 2019.
- [11] Z. Manna, Mathematical Theory of Computation, Dover Publications, 1974.
- [12] "Propositional calculus," [Online]. Available: https://en.wikipedia.org/wiki/Propositional_calculus.
- [13] "T-norm fuzzy logics," [Online]. Available: https://en.wikipedia.org/wiki/T-norm_fuzzy_logics.
- [14] "First-order logic," [Online]. Available: https://en.wikipedia.org/wiki/First-order_logic.
- [15] "Second-order logic," [Online]. Available: https://en.wikipedia.org/wiki/Second-order_logic.
- [16] "Prolog," [Online]. Available: <https://en.wikipedia.org/wiki/Prolog>.
- [17] "Logic Programming," [Online]. Available: https://en.wikipedia.org/wiki/Logic_programming.
- [18] OWL Working Group, "OWL Web Ontology Language Use Cases and Requirements," 2004. [Online]. Available: <https://www.w3.org/TR/webont-reg/#onto-def>.
- [19] OWL Working Group, "OWL 2 Web Ontology Language, 2nd ed., W3C Recommendation," 2012. [Online]. Available: <https://www.w3.org/TR/owl-ref/>.
- [20] P. Bellini, R. Mattolini, and P. Nesi, "Temporal Logics for Real-Time Systems Specification," ACM Computing Surveys, 2000.
- [21] "Temporal Logic," [Online]. Available: https://en.wikipedia.org/wiki/Temporal_logic.
- [22] "Knowledge engineering," [Online]. Available: https://en.wikipedia.org/wiki/Knowledge_engineering.
- [23] "Expert system," [Online]. Available: https://en.wikipedia.org/wiki/Expert_system.
- [24] "Rule-based system," [Online]. Available: https://en.wikipedia.org/wiki/Rule-based_system.
- [25] "Knowledge representation and reasoning," [Online]. Available: https://en.wikipedia.org/wiki/Knowledge_representation_and_reasoning.
- [26] "Ontology engineering," [Online]. Available: https://en.wikipedia.org/wiki/Ontology_engineering.
- [27] D. Koller and N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.
- [28] "Graphical model," [Online]. Available: https://en.wikipedia.org/wiki/Graphical_model.
- [29] T. R. Gruber, "A translation approach to portable," Knowledge Acquisition, 1993.
- [30] L. F. Sikos, Mastering Structured Data on the Semantic Web: from HTML5 Microdata to Linked Open Data, 2015.
- [31] I. Horrocks, U. Sattler, and F. Baader, "Chapter 3: Description Logics," in Foundations of Artificial Intelligence, 2007.
- [32] R. Arp et al., Building Ontologies with Basic Formal Ontology, The MIT Press, 2015.
- [33] J. Heleber et al., Semantic Web Programming, Wiley, 2009.
- [34] W. Wong et al., "Ontology learning from text: a look back and into the future," ACM Computing Surveys, Vol. 44, No. 4, Article 20, 2012.
- [35] "Semantic Web Stack," [Online]. Available: https://en.wikipedia.org/wiki/Semantic_Web_Stack.
- [36] N. Casellas, "Linked Legal Data: A SKOS Vocabulary for the Code of Federal Regulations," IOS Press.
- [37] S. Mandal et al., "Semantic Web Representations for Reasoning about Applicability and Satisfiability of Federal Regulations for Information Security," RELAW, Ottawa, 2015.
- [38] I. Sanya and E. Shehab, "A Framework for developing engineering design ontologies within the aerospace industry," International Journal of Production Research, vol. 53:8, pp. 2383-2409, 2015.
- [39] CRYSTAL - Critical System Engineering Acceleration, "State of the art for Healthcare ontology," vol. D407.010, 2013.
- [40] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes. Sensor fusion IV: control paradigms and data structures.," International Society for Optics and Photonics, vol. 1611, 1992.
- [41] S. M. LaValle and J. J. Kuffner Jr., "Randomized kinodynamic planning," The international journal of robotics research, vol. 20, no. 5, pp. 378-400, 2001.
- [42] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," The international journal of robotics research, vol. 30.7, pp. 846-894, 2011.

- [43] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24.6, pp. 381-395, 1981.
- [44] P. Jackson, Introduction to Artificial Intelligence, Dover Publications, 1985.
- [45] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in IEEE international conference on robotics and automation, 1985.
- [46] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE transactions on Systems Science and Cybernetics, vol. 4.2, pp. 100-107, 1968.
- [47] C. G. Harris and M. Stephens, "A combined corner and edge detector," in Alvey vision conference, 1988.
- [48] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," IBM Journal, pp. 211-229, 1959.
- [49] T. M. Mitchell, Machine Learning, McGraw-Hill, 1997.
- [50] K. P. Murphy, Machine Learning: A probabilistic Perspective, The MIT Press, 2012.
- [51] L. Breiman, "Statistical Modeling: The Two Cultures," Statistic Science, vol. 16(3), pp. 199-231, 2001.
- [52] D. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," IEEE Transactions on Evolutionary Computation, vol. 1 No 1, 1997.
- [53] D. H. Wolpert, "The Lack of A Priori Distinctions Between Learning Algorithms," Neural Computation, vol. 8, pp. 1341-1390, 1996.
- [54] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, 2009.
- [55] "Supervised learning," [Online]. Available: https://en.wikipedia.org/wiki/Supervised_learning.
- [56] S. Russell and P. Norvig, "Chapter 18. Learning from examples," in Artificial Intelligence: A modern approach, Prentice Hall, 2009, pp. 493-767.
- [57] A. A. Patel, Hands On Unsupervised Learning with Python, O'Reilly, 2019.
- [58] "Unsupervised learning," [Online]. Available: https://en.wikipedia.org/wiki/Unsupervised_learning.
- [59] "Semi-supervised learning," [Online]. Available: https://en.wikipedia.org/wiki/Semi-supervised_learning.
- [60] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., The MIT Press, 2018.
- [61] "Reinforcement learning," [Online]. Available: https://en.wikipedia.org/wiki/Reinforcement_learning.
- [62] "Meta learning," [Online]. Available: [https://en.wikipedia.org/wiki/Meta_learning_\(computer_science\)](https://en.wikipedia.org/wiki/Meta_learning_(computer_science)).
- [63] "Automated machine learning," [Online]. Available: https://en.wikipedia.org/wiki/Automated_machine_learning.
- [64] "Neuroevolution," [Online]. Available: <https://en.wikipedia.org/wiki/Neuroevolution>.
- [65] D. Grbic and S. Risi, "Towards continual reinforcement learning through evolutionary meta-learning," GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 119-120, 2019.
- [66] M. A. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
- [67] David Rumelhart et al., "Learning Internal Representations by Error Propagation," Defense Technical Information Center technical report, 1985.
- [68] Heng-Tze et al., "Wide & Deep Learning for Recommender Systems," in First Workshop on Deep Learning for Recommender Systems, 2006.
- [69] NASA, "Certification Considerations for Adaptive Systems," 2015.
- [70] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection: A Survey," ACM Computing Surveys, 2009.
- [71] SAE, "J3016 Taxonomy and Definitions for Terms related to Driving Automation Systems for On-Road Motor Vehicles," 2018.

2.2 Definitions

This set of definitions is meant to be a reference for the whole document.

ACCEPTANCE SCENARIO: A test or simulation procedure designed to gather evidence of a system's compliance with requirements and expectations.

ACCEPTANCE CRITERIA: The criteria applied for the complete set of acceptance scenarios.

ADAPTIVE SYSTEM: System that changes its behavior, based on an active feedback process in the presence of changes in the system or its environment. The environment can include hardware or software components of the computing platform, or the external surroundings in which the system operates (e.g., for airborne systems, changes to the physical structure of the aircraft, or in the weather). An adaptive system goal-seeks by iteratively updating parameters in response to environmental changes, but without resorting to predetermined values in look-up tables, or through defined calculations. This is the defining characteristic.

ARTIFICIAL INTELLIGENCE (AI): The theory and development of software-based systems able to perform tasks that have hitherto been the province of human intelligence. Examples include visual perception, speech recognition, decision-making, customer support and anomaly detection. John McCarthy, who coined the term in 1956, defines it as "the science and engineering of making intelligent machines."

ARTIFICIAL GENERAL INTELLIGENCE (AGI): Also known as general or strong AI, Artificial General Intelligence refers to an artificial system able to perform any intellectual task that can be performed by a typical human. The result would be a system considered to have the kind of general intelligence possessed by human beings and perhaps also qualities like consciousness.

ARTIFICIAL NARROW INTELLIGENCE (ANI): Artificial Narrow Intelligence, also known as weak AI, refers to the application of AI to solve narrow or specific problems. All current AI implementations are in this category.

ARTIFICIAL SUPER INTELLIGENCE (ASI): Artificial Super Intelligence research considers the possibility of systems that, given their superior processing and storage capabilities when compared to the human brain, could potentially exhibit super-human levels of intelligence. How such systems could be created, and how they would be evaluated is a matter of research and philosophical debate.

ARTIFICIAL NEURAL NETWORK (ANN): Neural Networks are algorithms, modelled loosely on the workings of the biological brain, where neurons are "fired" with sufficient stimuli. Typically, an ANN is "trained" with example data, in which the expected outputs are defined for a given input dataset. Parameters are adjusted based on the deviation from expected outcomes.

In "classical" ANNs, data is transmitted in one direction only, but there are two common architectures in which data is bi-directional, and these are better suited to certain classes of problem:

- **Convolutional Neural Network (CNN):** A type of Neural Network for processing data that has a known grid-like topology. CNNs use a specialized kind of linear operation in place of general matrix multiplication in at least one of their layers, which make them well suited to analytical applications, such as predictive maintenance.
- **Recurrent Neural Network (RNN):** A type of Neural Network that involves directed cycles in memory. One aspect of recurrent neural networks is the ability to build on earlier types of networks with fixed-size input vectors and output vectors, which make them well suited to operational applications, such as a self-driving car or self-flying airplane.

ATTRIBUTE: Information representing a property or characteristic of a subject as recorded in structured data, such as a table, spreadsheet, or database. In a database context, attributes are commonly referred to as fields, whereas in a spreadsheet or table context, attributes are commonly referred to as columns. For instance, structured data representing a person may have attributes such as name or age.

AUTOMATION: The use of control systems and information technologies reducing the need for human input, typically for repetitive tasks.

AUTONOMOUS [4]: Operations of an unmanned system wherein the system receives its mission from the human or agent and accomplishes that mission with or without further human-robot interaction (HRI). The level of HRI, along with other factors such as mission complexity, and environmental difficulty, determine the level of autonomy for the unmanned system. Finer-grained autonomy level designations can also be applied to the tasks, lower in scope than mission.

- **Fully Autonomous:** A mode of UMS operations wherein the UMS is expected to accomplish its mission, within a defined scope, without human intervention.
- **Semi-Autonomous:** A mode of UMS operations wherein the human operator and/or the UMS plan(s) and conduct(s) a mission and requires various levels of HRI.

For the automotive industry, six levels of autonomy are defined (refer to SAE J3016 [4]).

AUTONOMY: The ability to perform one or more tasks in a changing environment following a decision-making process without input by a human.

BIAS (MACHINE LEARNING): (1) An error deriving from erroneous assumptions in the learning process. High bias can cause an algorithm to miss the relevant relations between attributes and target outputs (known as underfitting). (2) A parameter within neural network that transforms input data. The inputs get modified by a bias value.

BIAS (STATISTICS): A feature of a statistical technique or of its results whereby the expected value of the results differs from the true underlying quantitative parameter being estimated.

BIG DATA: A discipline that specializes in dealing with the analysis of very large amounts of data, with high velocity (high speed of data processing.) The data may come from a wide variety of sources (sensors, images, texts, etc.) in a wide variety of formats, including unstructured formats, such as free text.

CLASSIFICATION (MACHINE LEARNING): A method that maps input data to one of a number of discrete classes.

CLUSTERING: A method that identifies similarities between features of the data and groups data items into clusters with similar features. It is a common example of unsupervised learning.

COMMERCIAL-OFF-THE-SHELF (COTS): Refers to generally available software and hardware developed for a broad market, rather than hardware or software designed specifically for aerospace applications. Special rules and regulations often apply when leveraging COTS hardware or software for safety critical applications, as such hardware or software may not have been developed in accordance with aerospace industry standards.

DATA CLEANSING: Identification and removal of errors and duplicate data to create a reliable dataset. This improves the quality of the training data for analytics.

DATA DRIFT: Phenomenon in which a Machine Learning system encounters data values during operation that were not used in training the underlying model. The predictive capabilities of the system are compromised as in-service data values diverge from those used in model training; for instance, the performance of an aircraft engine prognostics system trained with data from warm weather data, may degrade if the deployed system is used in cold weather over sustained periods.

DATA-DRIVEN AI: An approach to AI development focused on building a system that can output appropriate responses based on having learned from a large number of examples.

DATA SCIENCE [3]: A broad field that refers to the collective processes, theories, concepts, tools, and technologies that enable the extraction of information and analysis to acquire knowledge from that information.

DATASET (MACHINE LEARNING): The sample of data used for various development phases of the algorithm: i.e., training, validation, and test.

- **Training Dataset:** Data that is input to a Machine Learning model in order to establish its behavior.
- **Validation/Development Dataset:** Used to tune some hyperparameters of a model (e.g., number of hidden layers, learning rate, number of neurons per layer).

- **Test Dataset:** Used to assess the performance of a model, independent of the training dataset.

NOTE: These terms are not used consistently in the ML community.

DECISION MAKING: The selection of a course of action based on available information.

DEEP LEARNING (DL): A specific type of Machine Learning based on the use of large (deep) neural networks to learn abstract representations of the input data using multiple layers.

DETERMINISTIC SYSTEM: A system for which no randomness is involved in the development of future states. A deterministic system will always produce the same output given the same input (including environmental conditions) and initial state.

DERIVED REQUIREMENTS: Requirements which are not directly traceable to higher level requirements and/or specify behavior beyond that specified by the higher level requirements.

DOMAIN: Specific field of knowledge or expertise.

EXPECTED PROPERTY: The expected functional or performance property of the system outputs for each acceptance scenario.

EXPERT SYSTEM: A knowledge-based system that provides for solving problems in a domain or application area by drawing inferences from a knowledge base developed from human expertise. The term "expert system" is sometimes used synonymously with "knowledge-based system" but should be taken to emphasize expert knowledge. Some expert systems can improve their knowledge base and develop new inference rules based on their experience with previous problems. Expert systems usually rely on explicit, symbolic encoding of knowledge.

EXPLAINABILITY: The extent to which humans can understand, interpret, and account for the causality of an AI system or algorithm: why a particular output is produced for a given set of inputs. (Note: In this document, we use explainability in a broad manner to cover both explainability, as in a system's ability to show its work, and interpretability, as in a human's ability to logically understand what is being explained.)

EXPLAINABLE AI: An emerging field of study that aims to make the workings of AI systems transparent, such that humans can see process and decision flows. Note, that this may not be the same as having AI systems be logically understood by humans, although the goal of many Explainable AI projects is often aligned to that outcome. See definition for INTERPRETABLE AI.

FORMAL METHODS: A collection of mathematically rigorous techniques that can be applied to the specification, development and verification of software or hardware. A major benefit is a reduced need for human inspection and testing, for verification and validation, relying instead on unequivocal mathematical proofs of system properties. Challenges include the high level of required expertise, sparse tool support, and the limited applicability to all types of problem.

FUNCTION: The intended behavior of a product based on a defined set of requirements regardless of implementation.

GENETIC LEARNING [6]: An approach to Machine Learning based on an iterative classification algorithm which selects pairs of classifiers according to strength, and then applies genetic operators to the pairs to create offspring. The strongest offspring replace the weakest in order to generate new, plausible classifiers when the prior classifiers prove inadequate. The term "genetic" comes from the field of natural genetics, where it is linked to heredity governed by genes.

HYPERPARAMETER (MACHINE LEARNING) [8]: A setting that is used to control the behavior of a learning algorithm (e.g., number of hidden layers, learning rate, number of neurons per layer). Note: the values of hyperparameters are not adapted by the learning algorithm.

INFERENCE (MACHINE LEARNING): The process of a Machine Learning model computing an output, based on input data. The concept of inference originates from the field of Logic and is used extensively in Symbolic AI. The dictionary definition is "a conclusion or opinion that is formed because of known facts or evidence." See also "rule of inference" in [14] or "inference engine" in [25].

See also the related definition of "Training."

INTERNET OF THINGS (IoT): The network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment.

INTERPRETABLE AI: The quality of an AI system to explain process and decision flows in such a manner that humans can logically decompose and understand the system's output and decision making. With Interpretable AI, predictions can be logically justified, errors can be traced to their source in logical fashion, and a level of confidence in outcomes can be ascertained.

ITEM: A hardware or software element having bounded and well-defined interfaces.

KNOWLEDGE: A collection of facts, events, beliefs, and rules organized for systematic use.

KNOWLEDGE ACQUISITION: The process of locating, collecting, and refining knowledge and converting it into a form that can be further processed by a knowledge-based system. Knowledge acquisition normally implies the intervention of a knowledge engineer, but it is also an important component of Machine Learning. [6]

KNOWLEDGE BASE; K-BASE [6]: A database that contains inference rules and information about human experience and expertise in a domain. In self-improving systems, the knowledge base additionally contains information resulting from the solution of previously encountered problems.

KNOWLEDGE ENGINEERING [6]: The discipline concerned with acquiring knowledge from domain experts and other knowledge sources and incorporating it into a knowledge base. The term "knowledge engineering" sometimes refers particularly to the art of designing, building, and maintaining expert systems and other knowledge-based systems.

KNOWLEDGE-BASED SYSTEM: Information processing system that provides for solving problems in a domain or application area by drawing inferences from a knowledge base. The term "knowledge-based system" is sometimes used synonymously with "expert system," which is usually restricted to expert knowledge.

MACHINE LEARNING (ML): The branch of AI concerned with the development of algorithms that allow computers to evolve behaviors based on observing data and making inferences on these data.

Machine Learning strategies include four methods: [1][2]

- **Supervised Learning:** The process of learning a function that maps an input to an output based on labelled training dataset.
- **Unsupervised Learning:** The process of learning a function from a non-labelled dataset, by adapting the model to increase accuracy of the algorithm based on a given cost function.
- **Reinforcement Learning:** The process of learning in which the algorithm rewards positive results and punishes for a negative result, enabling it to improve over time. The learning system is called an agent.
- **Semi-Supervised Learning:** The process of learning in which the algorithm is capable of learning from data that is partially labelled.

MACHINE LEARNING MODEL: A parameterized function that maps inputs to outputs. The parameters are determined during the training process.

NATURAL LANGUAGE PROCESSING (NLP): Natural Language Processing (NLP) is a sub-field of Artificial Intelligence, concerned with computational techniques for analyzing, representing, and processing natural (human) language texts and voices, at one or more levels of linguistic analysis, such as phonetics, syntax, semantics and discourse, for the purpose of achieving human-like language abilities for a range of applications, including natural language understanding and generation, and speech recognition and generation.

OPERATIONAL DESIGN DOMAIN (ODD): Description of the specific operating domain(s) in which an automated function or system is designed to properly operate, including but not limited to operational aspects, environmental conditions, and other domain constraints.

ONLINE LEARNING ALGORITHM [2]: Represents a class of learning algorithms that learn to sequentially optimize predictive models over a stream of data instances while performing in the run-time environment during operations. The on-the-fly learning makes online learning highly scalable and memory efficient and differentiates it from batch learning or offline learning.

OPTIMIZATION: Methods that find the best available values within a function or distribution according to a defined cost function.

PREDICTABILITY: The degree to which a correct forecast of a system's state can be made quantitatively.

REGRESSION: A set of statistical techniques to postulate the mathematical relationship of one or more dependent variables to one or more independent variables and use such a relationship for statistical inference or prediction.

REINFORCEMENT LEARNING (MACHINE LEARNING): See Machine Learning definition.

ROBUSTNESS (SYSTEM): The extent to which the system handles invalid inputs properly, ability of a system to provide stable outputs with perturbations in the inputs.

ROBUSTNESS (SOFTWARE): The extent to which software can continue to operate correctly despite abnormal inputs and conditions.

ROBUSTNESS (MACHINE LEARNING MODEL): Ability of a model to generalize the outputs, for an input varying in a region of the state space compatible with the training range.

RULE-BASED SYSTEM; PRODUCTION SYSTEM: A knowledge-based system that draws inferences by applying a set of if-then rules to a set of facts following given procedures.

SEMANTIC NETWORK; SEMANTIC NET: A concept-based knowledge representation in which objects or states appear as nodes connected with links that indicate the relationships between various nodes.

SEMI-SUPERVISED LEARNING: See Machine Learning definition.

SUPERVISED LEARNING: See Machine Learning definition.

SYMBOLIC AI [3]: In contrast to Data-Driven AI, attempts to capture knowledge and derive decisions through explicit representation and rules.

SYSTEM: A combination of inter-related elements arranged to perform specific functions.

TESTING: The process of exercising a system or component to verify that it satisfies specified requirements.

TRAINING (MACHINE LEARNING): The process of optimizing the parameters of a Machine Learning model, given a dataset and a task to achieve on that dataset.

TRUSTWORTHINESS: Set of three qualities of a system that should be satisfied throughout its entire life cycle: it should be lawful, complying with all applicable laws and regulations, it should be ethical, ensuring adherence to ethical principles and values, and it should be robust, both from a technical and social perspective. [10]

UNMANNED SYSTEM (UMS): An electro-mechanical system, with no human operator aboard, that is able to exert its power to perform designed missions. May be mobile or stationary. Includes categories of unmanned ground vehicles (UGV), unmanned aerial vehicles (UAV), unmanned underwater vehicles (UUV), unmanned surface vehicles (USV), unattended munitions (UM), and unattended ground sensors (UGS). Missiles, rockets, and their submunitions, and artillery are not considered unmanned systems.

UNSUPERVISED LEARNING: See Machine Learning definition.

VALIDATION: The process of determining that requirements are both correct and complete with respect to representing larger goals and objectives.

VARIANCE (MACHINE LEARNING): An error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).

VARIANCE (STATISTICS): A standard calculation in Statistics that indicates the “spread” of a data distribution, whilst reducing the sensitivity to outliers. Essentially, it tells us how far a set of numbers are spread out from their average value. More formally, variance is the expectation of the squared deviation of a random variable from its mean.

VERIFICATION: The evaluation of the outputs of a process to ensure correctness and consistency with respect to the inputs and standards provided to that process.

WEIGHT (MACHINE LEARNING): A parameter within a Neural Network that transforms input data. The inputs get multiplied by a weight value.

- Positive weight → excitatory relation
- Negative weight → inhibitory relation

NOTE: Both bias and weight are learnable parameters inside the network.

WEIGHT (SYMBOLIC AI): A parameter within Semantic Networks which indicates the strength of the relationship between the nodes.

WELL FORMULATED FORMULA (wff): A statement whose validity can be determined by mathematical logic.

3. CLASSIFICATION OF AI TECHNIQUES

This section offers a common knowledge base of AI terminology as a foundation for the remaining sections. We provide a taxonomy of AI concepts and techniques based on three major paradigms: Symbolic AI, Numerical Analysis, and Machine Learning. The section concludes with a discussion of a general workflow for developing ML systems.

Figure 1 shows a classification of AI techniques adopted by the SAE G-34/WG-114 committee, and those techniques relevant to the statement of concerns document are detailed in 3.1.

Main Classification:

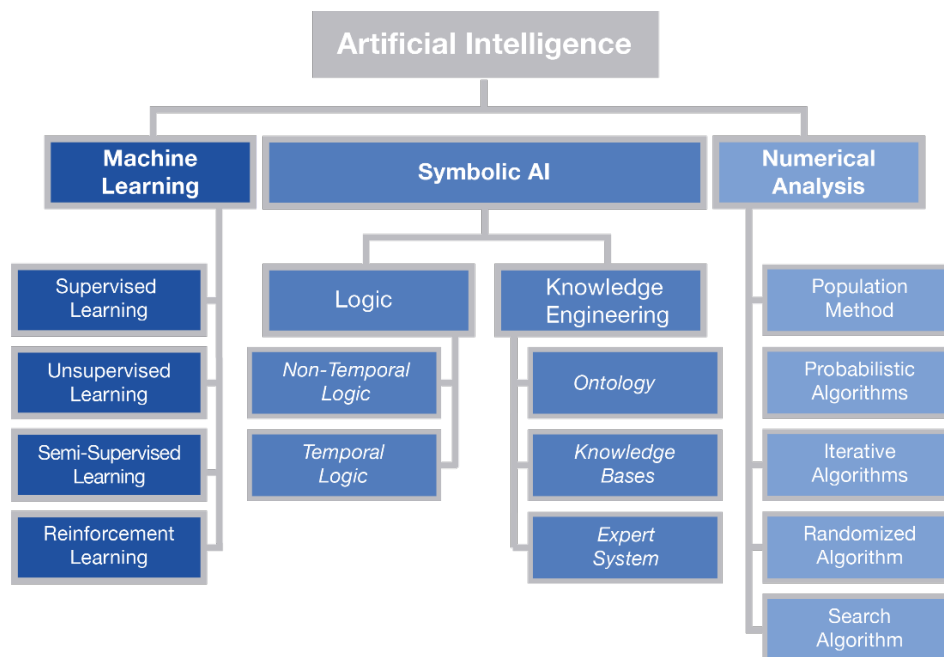


Figure 1 - AI classification

3.1 Symbolic AI

Symbolic AI refers to the collection of methods that attempt to capture and encode human knowledge, for the purpose of machine understanding and processing. Solutions are based on human-readable representations of problems, where real-world objects, along with their characteristics, relationships, and interactions, are represented by symbols. An aircraft, for example, could be represented by the textual string “aircraft.” Although this approach requires no model training, no massive amounts of data, and no “guesswork,” its main limitation lies in the difficulty of codifying the real worlds, with all its complexity and nuances. Today’s knowledge-based AI agents occupy only a partially described universe.

3.1.1 Logic

AI and Computer Science evolved together over the second half of the 20th century and, since computers are fundamentally Boolean Logic processing machines, it is natural to apply the principles and techniques of Logic to solve problems. The approach is to build propositional sentences, using a constrained grammar, that express some belief (“Truth”) about the domain of interest, with no ambiguity. A modern computer can assess the truth of millions of propositions per second, to make decisions, plan, and act.

The logic branch of Symbolic AI is composed of several classes of Formal Methods. They allow the representation of logical arguments (a.k.a. statements) in terms of well-formed formulas (wffs), whose validity can be determined by mathematical logic. A wff is said to be valid if its tautology can be demonstrated for all possible interpretations [11]. An interpretation consists of an assignment of truth values to the wff variable symbols.

3.1.1.1 Knowledge Engineering

Knowledge Engineering [22] refers to all technical, scientific, and social aspects involved in building, maintaining, and using knowledge-based systems. This broad field includes long-standing concepts such as expert systems [23] and rule-based systems [24], as well as more recent techniques such as ontology engineering [25], [26] and probabilistic graphical models [27], [28]. A common trait of knowledge engineering solutions is the hard coding of knowledge using formal languages and their processing using formal methods.

3.1.1.1.1 Ontologies

Ontologies [18], [29], [30], [31] are data structures that model concepts, roles, and individuals, and their relationships. Ontologies can store conceptual or incomplete information and use reasoning over description logic to infer missing relationships and attributes.

Ontologies [32] capture domain knowledge by organizing concepts, relationships, and constraints organized in a web of statements named the Semantic Web [33]. The ontology information model is designed to facilitate easy data sharing, reuse, and interoperability across multiple scientific and engineering domains. The Semantic Web accomplishes this through two critical principles: decoupling the knowledge model from the application and integrating knowledge models through reuse and extension [27]. Many ontologies can be defined for any single domain since their creation is usually context driven in terms of scope, abstraction-level, granularity, properties, and intent.

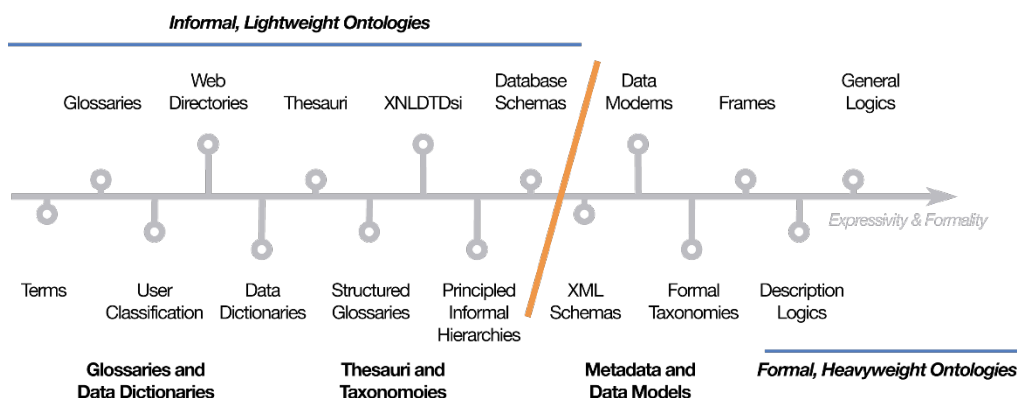


Figure 2 - The spectrum of ontology kinds [34]

3.1.1.1.2 Knowledge Bases

A Knowledge Base (KB) [27] is a software component that represents a collection of facts or statements that is ontologically described, processed, and accessed in a Semantic Web application. Entailments inferred by reasoners combined with user/application asserted statements in the KB can refer to generic concepts or specific individuals (instance data). Reasoners evaluate the assertions in the underlying data structures and verify consistency of ontology concepts and their mutual relationship. The use of ontologies allows a KB to incorporate new ontologies and instance data incrementally as the need arises.

3.1.1.1.3 Expert Systems

In artificial intelligence, expert systems are designed to solve complex problems by reasoning through bodies of knowledge, represented mainly as “if-then” rules (production rules) rather than through conventional procedural code. An expert system usually has two core components: a knowledge base and an inference engine. The knowledge base is an organized collection of facts and production rules about the system’s domain. Facts are frequently acquired from human experts through interviews and observations. The inference engine applies the rules to the known facts in the knowledge base to possibly deduce new facts in order to emulate the decision-making ability of a human expert. Inference engines can also include explanation and debugging abilities. Typical tasks for expert systems involve classification, diagnosis, monitoring, scheduling, and planning for specialized technology domains.

3.1.2 Numerical Analysis

This section covers several common algorithmic approaches based on non-data driven techniques used in a wide range of applications including Optimization, Mapping, and State Estimation, in applications such as robotics and autonomous vehicles.

3.1.2.1 Algorithmic Approaches

3.1.2.1.1 Iterative Algorithms

Iterative approaches can be described as taking an initial guess at a solution, and repeatedly running a procedure (an iteration) to refine that guess to get closer and closer to the acceptable solution. The iterations typically continue until either a maximum number of iterations are reached, or a certain fitness of the solution is found. Depending on the problem, the fitness of the solution may be a measure of the proximity to the true solution, or it may be a heuristic measurement.

3.1.2.1.2 Randomized Algorithms

Randomized algorithms explicitly use random numbers as one of their inputs as a method to converge to a true solution. These algorithms generally use random inputs as samples from a large action or state space. As the number of samples increases, these algorithms generally then either heuristically or provably converge to the correct solution.

3.1.2.1.3 Population Methods

Population methods involve self-affecting, multi-part populations [46]. A mathematical system that “affects itself” is typically composed of at least two parts that produce and accept feedback to themselves, that is, where the action of any part affects the other parts, which in turn affect the original part [44].

In ANNs, for example, population-based training is a hyperparameter optimization technique, similar to genetic algorithms, that learns from a schedule of hyperparameters rather than fixed values.

Population-based methods mimic the genetic search of Natural Selection in the biological world. Instead of working with a single candidate solution, the designer works with a population of candidates to explore the solution space.

Genetic (Evolution) Algorithms have a flavor of “bottom-up” or “emergent” problem solving.

3.1.2.1.4 Probabilistic Algorithms

Probabilistic algorithms describe a general class of algorithms that operate directly on probability distribution functions (PDF). Rather than computing a value directly, they compute the probability that a random variable takes any given value. These methods can be useful in developing systems with inputs that have an associated uncertainty (such as sensor data), as they allow for passing that uncertainty through the full system. Many probabilistic algorithms fall into the category of Bayesian Inference, which make use of Bayes' Rule to update a prior distribution with new information.

3.1.2.1.5 Search Algorithms

Search algorithms handle retrieving data or finding connections in a given data structure. This includes applications such as finding an optimal path between two nodes on a graph or identifying a subset of items within the structure that fulfills certain criteria.

3.1.2.1.6 Direct Approaches

Direct approaches can best be described by not being iterative approaches; they constitute a predetermined set of operations independent of the input data. Direct approaches cover a wide range of algorithms from algorithms with exact solutions (such as linear systems or Kalman filters) to algorithms that simply change the representation of data (such as convolutions). Many algorithms in both traditional and learning based computer vision fall in the category of direct approaches.

3.1.3 Machine Learning (ML)

The ML field is crowded with multiple learning strategies and algorithm families with no discernible advantage among alternatives if one makes absolutely no assumption about the training data (this is known in the AI R&D community as the No Free Lunch theorem [8], [52], [53]). The generally accepted classification of ML algorithms is: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning.

3.1.3.1 Supervised Learning

With Supervised Learning, the main goal is to train a model with labelled data, enabling the model to make predictions on new unseen data. The training data comprises samples, where the desired output (labels) are known and correct, for a given input.

In Classification problems, the model assigns a categorical class label to each sample. An example of Classification is image recognition, where physical objects can be classified as cars, people, signs, buildings, and so on.

A second type of Supervised Learning is the prediction of continuous outcomes, referred to as Regression. An example of regression is predicting the remaining useful life of an engine hardware component, given explanatory variables such as number of flight cycles, hours of operation, and operating temperature distribution [50], [54], [55], [56].

3.1.3.2 Unsupervised Learning

In contrast to Supervised Learning, where the right answer is already known for a number of samples, Unsupervised Learning deals with unlabeled datasets, or data of an unknown structure. Unsupervised algorithms learn underlying structures in the training set and use these patterns to make predictions. In Clustering, a common Unsupervised Learning technique, an amorphous pile of information can be sorted into meaningful subcategories (clusters), using features that may not be immediately discernable by humans. Typical unsupervised learning tasks include anomaly detection, visualization, dimensionality reduction, and association rule learning [50], [57], [58], [56].

3.1.3.3 Semi-Supervised Learning

The ML algorithm is capable of learning from data that is partially labelled. Most semi-supervised learning algorithms use unsupervised learning approaches to improve supervised learning solutions in dealing with problems such as insufficient labelled data, curse of dimensionality, feature engineering, outliers, and data drift.

3.1.3.4 Reinforcement Learning

In Reinforcement Learning, the goal is to produce a Machine Learning algorithm (agent) that improves its performance interacting with its environment and observing changes brought about by the agent's own actions. The learning objective is to determine an action-selection policy that maximizes a numeric reward signal. The agent must discover which actions yield the most reward by trying them. The trial-and-error search and the learner's need to consider delayed rewards are the two most important distinguishing features of Reinforcement Learning.

A classic example of Reinforcement Learning is where the objective, such as the safe landing of a spacecraft without crashing, is well defined. The algorithm uses repeated attempts to determine the optimal set of maneuvers and fuel usage to best achieve the objective [60], [61], [56].

3.1.3.5 Machine Learning Bubble Chart

To summarize the types of learning and their applications, a bubble chart was designed. The intent was not to have an exhaustive list of applications for each type of learning, a task that is impossible considering the variety of possible applications of Machine Learning. On the contrary, this chart provides only a few examples of technical applications (smaller bubbles) for each of the types of learning (medium size bubbles).

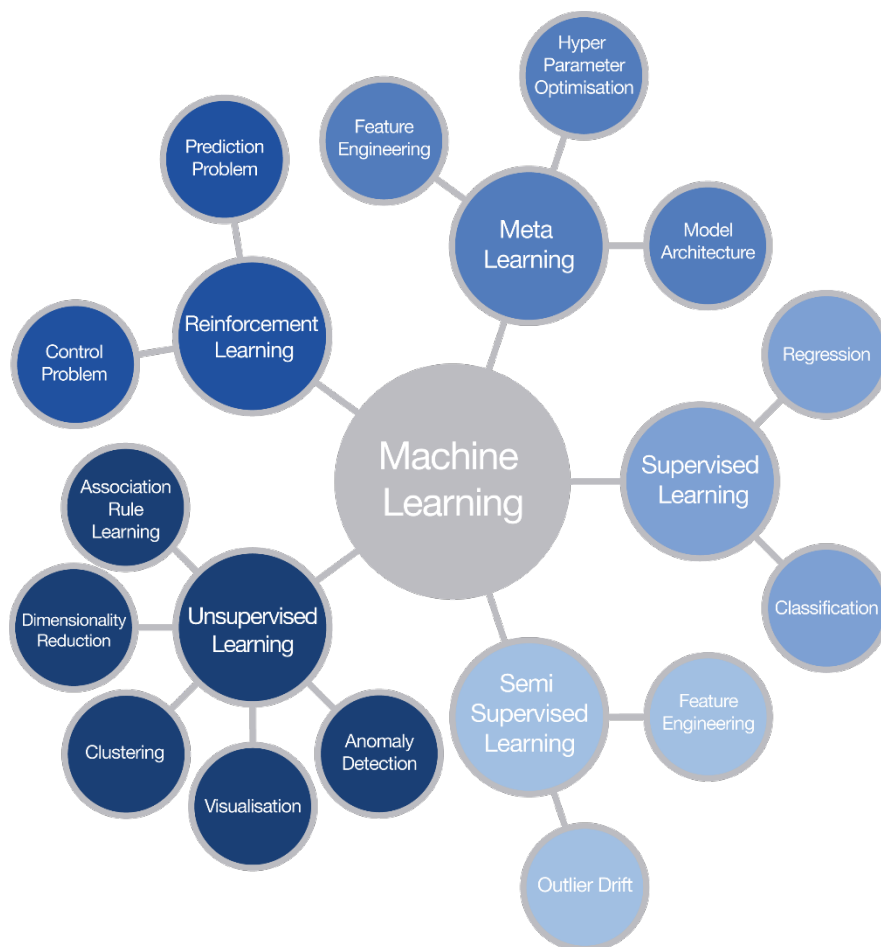


Figure 3 - Machine learning bubble chart

3.1.3.6 Neural Networks and Deep Learning

Artificial Neural Nets (ANNs) manifests from a Machine Learning modelling technique that was originally inspired by the networks of neurons in human brains, but gradually evolved apart from its biological analogy. Sufficient to say, the development of powerful graphical processing units (GPUs), availability of huge amounts of training data, improvements in ANN training algorithms and programming APIs, and a "virtuous circle of funding and progress" [5] have led to an accelerated evolution in the development of ANN-based products.

Original ANN architectures include the basic Perceptron and their layered combinations called Multi-Layer Perceptrons (MLPs). Fundamentally, an MLP is “just a mathematical function mapping some sets of input values to output values” [8]. Structurally, MLPs are dense networks of interconnecting layers of simpler perceptron units. The signal in MLPs flows only in one direction from the input layer towards the output layer passing sequentially through one or more hidden layers. The output of a unit in any given layer in an MLP network is a linear combination of the output of the units of the previous layer in the sequence. Training an MLP then consists in determining appropriate connection weights and respective biases terms for all units in the network. Nonlinear activation functions are interposed between consecutive layers to increase the expressiveness of the network.

The ANN is called a Deep Neural Network (DNN) when it has a deep stack of hidden layers, as opposed to Shallow Neural Networks (SNN) that have only a few hidden layers. There is no consensus about how much depth an ANN requires to be classified as a DNN. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, however, state in their seminal book [8] that “deep learning can be safely regarded as study of models that involve a greater amount of composition of either learned functions or learned concepts than traditional Machine Learning does.”

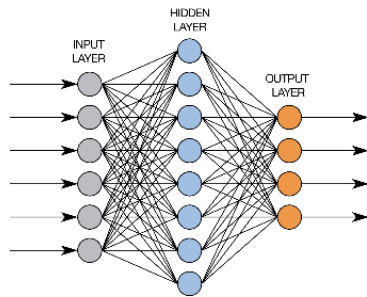


Figure 4 - Example of A SNN [66]

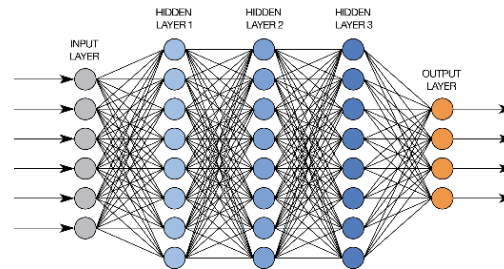


Figure 5 - Example of A DNN [66]

3.2 Machine Learning System Workflow

The workflow illustrated in Figure 6 is proposed to support the discussion in the following sections.

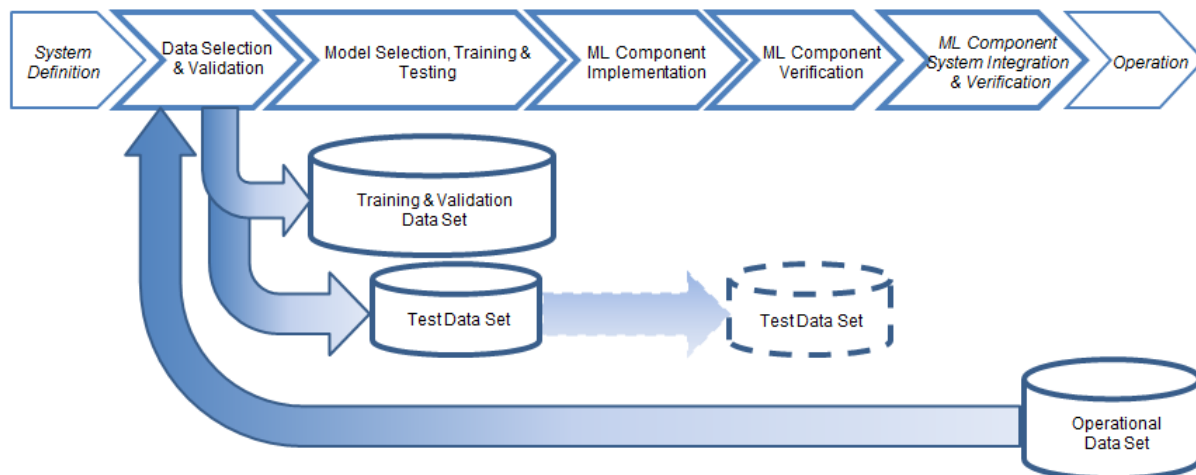


Figure 6 - Machine learning system workflow

The workflow comprises of the following phases:

System Definition:

The system requirements, allocated to the ML system, are captured and validated.

The system architecture is also defined at this stage and corresponding safety assessment and analyses are conducted to identify the relevant system safety and security requirements and objectives (quantitative, qualitative, or both) for the other phases.

Data Selection and Validation:

In this phase, the data should be selected and collected based on the specific problem domain. The data should contain a distribution of the data that includes the proper level of variances. The data is cleaned and/or transformed (using several techniques), with respect to specific expected quality attributes, and validated against requirements established in the previous phase.

Each data sample is allocated to either the training, validation, or test set.

Model Selection, Training, and Testing:

In this phase, the model is selected and trained with the Training dataset and then optimized with the Validation Dataset.

At the end of this phase the trained model is tested with the Test dataset, to assess the performance regarding the system requirements.

ML Sub-System Implementation:

In this phase the tested trained model is implemented (Design and HW/SW integration) into the target environment.

ML Sub-System Verification:

In this phase the correct implementation of the tested trained model is verified. The verification could be based on the test dataset in addition to the other test inputs established for the purpose of the implementation verification.

ML Sub-System Integration and Verification:

In this phase, the step by step integration within the overall system is performed and corresponding verification activities are conducted.

Operation:

In this phase, the overall system containing the ML system is released to field and operated. The behavior of the ML system is monitored, and the operating data are compiled into an operational dataset which is fed back to Data Selection and Validation to augment the existing datasets.

4. GAP ANALYSIS FROM EXISTING STANDARDS

4.1 Introduction

4.1.1 Description

There is a concern that existing development assurance standards may not be appropriate for AI/ML solutions. Existing development assurance standards require a predefined set of activities and demonstrations used later by certification authorities as evidences of compliance. There is a broad consensus the currently accepted means of compliance in use for systems, software and hardware fail to provide appropriate assurance for some specific of AI/ML techniques. These techniques use methodologies that are fundamentally different from the generic life-cycle assumed by the existing development assurance standards.

Although these techniques are not governed by a strict methodology and could be assessed in various ways, this section focuses on a predefined hypothetical development scenario. It does not intend to cover all possible scenarios and possibilities, but rather tries to highlight the main gaps identified in a factual way among existing development assurance standards in use today. It is important to note that this is not a guidance on how to certify a system using an AI sub-system.

Many published standards are considered in this section and some of them have been subject to a deeper analysis. The choice of standards assessed is led by the desire to cover at least system, software, and hardware aspect of ground and airborne development assurance standards.