Iris Groher, Thomas Vogel (Hrsg.): Software Engineering 2023 Workshops,
208 Digital Library, Gesellschaft für Informatik, 2023

Tool Qualification Aspects in ML-Based Airborne Systems Development

Konstantin Dmitriev[1] Fateh Kaakai[2] Mohamad Ibrahim[3] Umut Durak[4] Bill Potter[5] Florian Holzapfel[6]

Abstract: Machine Learning (ML) technology can provide the best results in many highly complex tasks such as computer vision and natural language processing and quickly evolving further. These unique ML capabilities and apparent potential can enable the next epoch of automation in airborne systems including single pilot or even autonomous operation of large commercial aircraft. The main problems to be solved towards ML deployment in commercial aviation are safety and certification, because there are several major incompatibilities between ML development aspects and traditional design assurance practices, in particular traceability and coverage verification issues. In this paper, we study the qualification aspects of tools used for development and verification of ML-based systems (ML tools) and propose mitigation measures for some known ML verification gaps through ML tools qualification. In particular, we review the DO-330 and DO-200B tool classification approach with respect to ML-specific workflows and propose to extend the tool qualification criteria for ML data management and ML model training tools.

Keywords: Machine Learning; Certification; Design Assurance; Tool Qualification

1 Introduction

Machine Learning (ML) technology is rapidly developing. By now it can provide the best results in many highly complex tasks such as computer vision and natural language processing comparing to traditional non-ML software [LBH15]. These unique ML capabilities and their growth potential can enable the next epoch of automation in aerospace systems including single pilot and even autonomous operation of large commercial aircraft. The main problem to be resolved towards the deployment of ML-based systems in commercial aviation is addressing certification issues because there are several major incompatibilities between ML development aspects and traditional design assurance practices. In particular, some

[1] Institute of Flight Systems Dynamics, Technische Universität München, German, Boltzmannstrasse 15, 85748 Garching, Germany konstantin.dmitriev@tum.de
[2] Thales, 3 avenue Charles Lindbergh, 94628 Rungis, France fateh.kaakai.e@thalesdigital.io
[3] Institute of Informatics, Technische Universität Clausthal, Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany mohamad.ibrahim@tu-clausthal.de
[4] Institute of Flight Systems, German Aerospace Center (DLR), Lilienthalplatz 7, 38108 Braunschweig, Germany umut.durak@dlr.de
[5] The MathWorks, Inc., 1 Apple Hill Drive, Natick, MA, USA bpotter@mathworks.com
[6] Institute of Flight Systems Dynamics, Technische Universität München, German, Boltzmannstrasse 15, 85748 Garching, Germany florian.holzapfel@tum.de

doi:10.18420/se2023-ws-19

fundamental verification techniques including traceability and structural coverage analysis are irrelevant to ML model verification [EU21, SA21].

These incompatibilities result in major gaps in verification processes and jeopardize the overall confidence in the safe operation of ML-based systems. Industry, academia, and aviation authorities are currently working to create new standards for ML-based systems certification, but this is rather a lengthy process and until new standards are released, the current practices must be followed [DSH21]

Current avionics certification standards such as RTCA DO-178C [RT11a] and DO-330 [RT11b] allow to reduce or to eliminate verification activities upon conditions that relevant tools are qualified for corresponding verification credit. In this work, we propose to extend this principle and consider the qualification of tools used for development of ML-based systems as a mean to mitigate some intrinsic verification gaps of ML development workflows. We also reconsider tool classification criteria for ML data management tools and ML training tools in order to address the discovered incompatibilities of ML workflows with existing industry standards.

The rest of this paper is structured as follows: in Section 2, we discuss the background of the topic and related work. Section 3 includes the overview of the ML tools in the context of generic ML development lifecycle. Then, we review the particular aspects of ML-specific tools and discuss the new qualification criteria for such tools in Section 4. Section 5 summarizes the work and outline suggestions for future work.

2 Background and Related Work

2.1 Design Assurance of ML-Based Airborne Systems

EASA "Concept Paper: First usable guidance for Level 1 machine learning applications"[EA21a] is one of the latest publications in a series of released documents on the approval of ML applications in the aviation domain which started with EASA AI Roadmap [EA20a], then EASA Concepts of Design Assurance for Neural Networks (CoDANN) I and II [EA20b, EA21b]. [7] This guidance introduces an initial set of objectives and anticipated means of compliance (MOC) for certification approval of ML applications in order to provide a basis for further refinement in certification projects and standardization bodies. It identifies four building blocks for ML-based system assurance:

1. **The trustworthiness analysis:** it acts as an interface between ethical and technical aspects of the framework.
2. **Learning assurance:** the objectives here address the learning process that represents the key difference from the traditional development paradigm.

[7] For full survey on ML regulatory frameworks, the reader can refer to [TDD22]

3. **Explainability:** it deals with making clear for humans how the results of ML-based application are concluded.

4. **Safety risk mitigation:** contains objectives to let the user address the residual risks that remained unhandled from the previous activities.

In addition to the ML Airworthiness framework and the objectives, EASA guidance presents a classification of ML applications, a W-shaped development process and a decomposition of the ML system. However, it does not address the aspects of tool qualification in ML workflows.

To create harmonized ML assurance guidelines, two standardization organizations, namely the Society of Automotive Engineers (SAE) and the European Organisation for Civil Aviation Equipment (EUROCAE), formed a joint international committee, SAE G-34 / EUROCAE WG-114, to tackle the task. The committee is working on the standard AS6983 "Process Standard for Development and Certification/Approval of Aeronautical Safety-Related Products Implementing AI". The authors are part of a group responsible for the tool qualification section of the standard. This work proposal might be included in the final version of AS6983.

In academia, a number of research group currently study different aspects of ML certification problem [Ma21, Hu20, Ta22, Vi21, To23]. However, we were not able to find during literature search significant information about qualification of ML tools in safety critical context.

2.2 Tool Qualification

In airborne software development, DO-330/ED-215 “Software Tool Qualification Considerations” [RT11b], one of the supplements of DO-178C/ED-12C Software Considerations in Airborne Systems and Equipment Certification"[RT11a], is the de facto standard for software tool qualification. Federal Aviation Administration (FAA) and the European Union Aviation Safety Agency (EASA) have recognized DO-330/ED-215 as an acceptable means of compliance in Advisory Circular AC-20-115D [FA22] and AMC 20-115D [EA] respectively. That made DO-330/ED-215 the primary guideline standard to obtain certification for software in combination with DO-178C/ED-12C. Other means are acceptable; however, proving that these other means comply with the regulations is a complicated and uncertain process. [8]

RTCA DO-178C/ED-12C defines a tool as “a computer program or a functional part thereof, used to help develop, transform, test, analyze, produce, or modify another program, its data, or its documentation”. The qualification processes start with evaluating the tool’s impact on the process workflow of the software lifecycle. DO-178C/ED-12C defines three criteria (Tool Impact) for the Tool under Qualification:

[8] For more details on tool qualification, the reader can refer to [ID21]

Tab. 1: DO-178C Tool Impact Criteria [RT11a]

a.	Criteria 1: A tool whose output is part of the airborne software and thus could insert an error.
b.	Criteria 2: A tool that automates verification process(es) and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of: 1. Verification process(es) other than that automated by the tool, or 2. Development process(es) that could have an impact on the airborne software.
c.	Criteria 3: A tool that, within the scope of its intended use, could fail to detect an error.

Criteria 1 corresponds to the Development Tools category from DO-178C, while Criteria 2 and 3 represent verification tools. Verification tools should be classified per Criteria 2 if they are used for eliminating or reducing other objectives or activities that are not directly related to the tool under qualification. Criteria 2 was introduced to prevent the breaching of the multilayered safety approach enforced by the standard.

DO-178C categorizes software into levels A through E, with A being the most severe "Catastrophic"category and E being the least severe "No Safety Effect"category. The Tool Qualification Level (TQL) is determined based on the tool's criteria and the criticality level of software developed using the tool. The following table depicts the mapping of tool qualification levels to different software levels and tool's criteria:

Tab. 2: Tool Qualification Level Determination [RT11a]

Software Level	**Criteria**		
	1	***2***	***3***
A	TQL-1	TQL-4	TQL-5
B	TQL-2	TQL-4	TQL-5
C	TQL-3	TQL-5	TQL-5
D	TQL-4	TQL-5	TQL-5

The determined TQL will dictate what objectives should be met in the qualification processes, which are:

1. Tool Planning Process
2. Tool Operational Process
3. Tool Development Process
4. Tool Verification Process
5. Integral Processes is done throughout the entire tool qualification lifecycle.

We will base our approach on the principles provided by DO-330/ED-215 and tailor the tool classification and levelling, considering the characteristics of software tools used for the ML-based development process. That is aligned with what has been done for RTCA DO-200B/EUROCAE ED-76A "Standards for Processing Aeronautical Data" [RT15].

DO-200B is similar to DO-178C, however, with a focus on databases and tools involved in the data generation and data error detection tools. DO-200B/ED-76A extended the tool qualification process described in DO-330/ED-215, introduced data quality characteristics' requirements and provided enhanced validation and verification activities to ensure data integrity. It applies not only to aircraft but to any aviation-related data that could affect the safety of the systems consuming them and to data not known at the time of certifying the system.

As defined by DO-200B/ED-76A, data quality characteristics include integrity, accuracy, completeness, resolution, traceability, timeliness and format. Those characteristics should be verified along the data path from supplier to customer. As much of the data processing is done through the use of tools, the standard handled the data tools and classified them as follows:

1. **Data Processing Tool:** the tools that have the ability to insert an error in the aeronautical data.
2. **Error Detection Tools:** the tools that could fail to detect an error in the aeronautical data

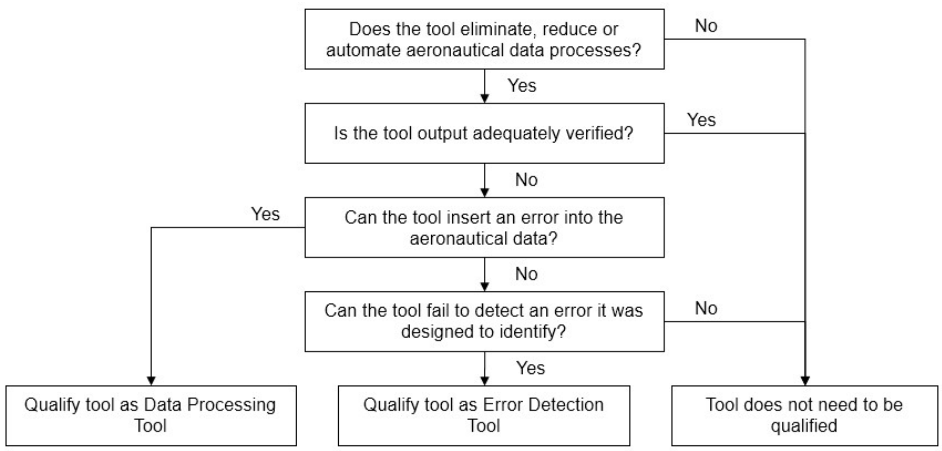

Fig. 1: Criteria for Tool Qualification [RT15]

The Committee that worked on DO-200B/ED-76A determined TQL-1 and TQL-2 to be irrelevant in the aeronautical data processing domain. Thus, the standard adapted the TQL determination from DO-178C/ED-12C to suit data processing tools qualification. The design Assurance Level (DAL) concept is changed to Data Processing Assurance level (DPAL) with only three levels as shown in Tab. 3.

Tab. 3: FAILURE CONDITION CATEGORIES [RT15]

<table>
<tr><th>Failure Condition Category</th><th>Design Assurance Level</th><th>Data Process Assurance Level</th></tr>
<tr><td>Catastrophic</td><td>A</td><td rowspan="2">1</td></tr>
<tr><td>Hazardous/Severe-Major</td><td>B</td></tr>
<tr><td>Major</td><td>C</td><td rowspan="2">2</td></tr>
<tr><td>Minor</td><td>D</td></tr>
<tr><td>No Safety Effect</td><td>E</td><td>3</td></tr>
</table>

As a result, the tool qualification level determination became as shown in Tab. 4. The rationale behind this adaptation is tailoring the objectives of DO-330/ED-215 to include only the applicable ones related to data processing tools. That is done by adjusting the level determination process to fit the peculiarities of data processing.

Tab. 4: Tool Qualification Level Determination [RT15]

DPAL	**Data Processing Tool**	**Error Detection Tool**
1	TQL-3	TQL-5
2	TQL-4	TQL-5
3	Not required	Not required

Concerning the aspects of tools qualification in the context of ML-enabled safety-critical system, we found that there is generally a lack of research work in this domain, as was also noted in the previous section.

3 Machine Learning Tools Overview

The EUROCAE/SAE Machine Learning Development Lifecycle (MLDL) is one of the key results of the joint working group EUROCAE WG114 / SAE G-34, which was constituted to address certification/approval challenges of Machine learning technologies as detailed in AIR6988 (statement of concerns) [EU21, SA21]. The MLDL has eleven (11) main process activities supported by several types of tools as depicted in Fig.2 and described below. The activities in Fig.2 are shown sequential for simplicity but generally iterative due to incremental nature of development and verification.

1. **ML requirements specification:** this process activity is similar to the requirements process activities described RTCA DO-178C [RT11a]. From the outputs of the system processes, the ML requirements specification develops requirements classified

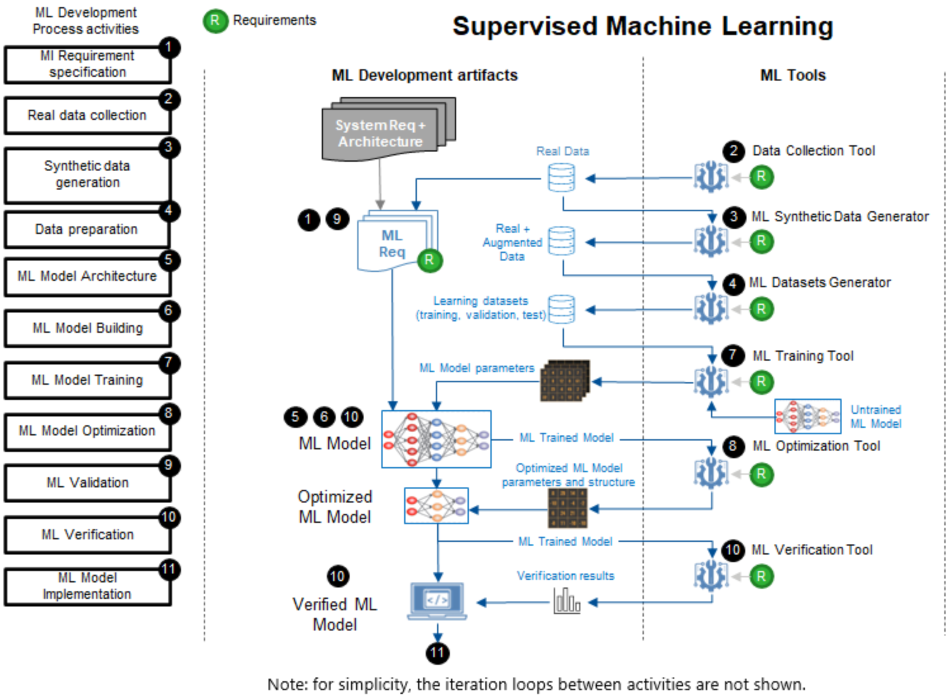

Fig. 2: Typical Supervised Machine Learning Development Lifecycle

into three (3) categories: (i) the ML functional requirements (equivalent to high-level requirements in DO-178C [RT11a]), (ii) ML data requirements (equivalent to low-level requirements in DO-178C [RT11a]), and (iii) ML model requirements (equivalent to low-level requirements in DO-178C [RT11a]).

2. **Data collection:** collect data from the identified sources with the desirable characteristics identified in the ML requirements (e.g., completeness, representativeness, lack of undesirable bias, etc.)

3. **Synthetic data generation:** generate artificial data from real data and/or from a simulation tool to augment the volume of data to be used in the learning activity.

4. **Data preparation:** clean the data, define input features for the ML model, label the data and allocate the data to three datasets: training, validation, and test datasets. The test dataset should not be used to build, train, or optimize the model.

5. **ML Model architecture development:** define the ML model logical architecture including the breakdown into ML model elements, the description of these individ-

ual ML model elements, and how they should be integrated to comply with ML requirements.

6. **ML Model building:** develop the analytical/algorithmic form of the ML model and identify the values of the ML model hyper-parameters to comply with the applicable ML requirements.
7. **ML Model training:** compute the optimal ML Model parameters using the appropriate ML training algorithm and the training/validation datasets to comply with the applicable ML requirements.
8. **ML Model optimization:** optimize the trained candidate ML Model to achieve expected performance (e.g., memory and throughput) as specified in the ML model requirements (e.g., hyperparameter search and quantization).
9. **ML Validation:** provide assurance that all levels of ML requirements are accurate and consistent, compatible and traceable between different levels, compatible with the target computer, verifiable, and conform to standards.
10. **ML Verification:** ascertain that ML implementation artifacts satisfy the validated ML requirements through (i) the verification of ML data specified quality attributes, (ii) the verification of the ML Model, its architecture, and description, and (iii) the verification of verification procedures.
11. **ML Model Implementation:** implement on the target computer the trained and optimized ML Model to meet the ML Model Description (specification) and implementation/performance constraints coming from system requirements.

In Fig.2, the outcomes of the MLDL process activities are called development artifacts and they contribute to the concept of Learning Assurance [EA21a]. The diagram in this figure is divided into two (2) parts: (i) on the left-hand side, the development artifacts are depicted and they are linked with the MLDL process activities that are done by the designer, and (ii) on the right-hand side, the typical ML tools are depicted with a reference to the MLDL process activities that are automated through these tools (the intended functions of these tools are specified by ML requirements). Some development artifacts are produced involving both the designer and the tools that support the activity. A common example is labeled datasets. The labeling activity (which is part of activity 4 - Data Preparation) is done in general by humans supported by tools. It is possible to observe in Fig.2 that supervised machine learning is highly automated with many tools contributing to most of the MLDL process activities. Tool qualification considerations are therefore one of the key issues in the overall ML assurance strategy.

4 Qualification aspects of ML-specific tools

In this section, we review the traditional tool qualification approach described in [RT11c] and DO-200B [RT15] with respect to the specific aspects of ML workflows and show that this approach is not fully relevant for some types of ML tools and should be adjusted to address certain unique aspects of ML workflows. We consider two following distinctive groups of ML-specific tools:

- ML data management tools
- ML model development tools

Other types of ML tools described in section 3 have no distinct properties and the standard DO-178C criteria and level mapping can be applied.

4.1 ML Workflow Assumptions

In our analysis we limit the review to the generic ML workflows for non-adaptive supervised learning systems that is currently one of the most common ML paradigm considered as a most probable candidate for the first airworthiness certification approval [EA21a, DSH21, DSH22a, DSH22b]. In such workflows, ML model can be considered as a representation of software design in terms of DO-178C [RT11a] and DO-331 [RT11c] definitions (because source code can be directly generated from ML model) therefore relevant verification objectives per existing or presumed future standards should be achieved for ML models. ML training tool qualification can support fulfillment of these verification objectives in particular for the known traceability verification gaps.

ML training datasets from which the ML model is produced in such workflows are ML-specific artifacts with unique properties that have no apparent counterpart in traditional software development workflows and need to specifically address with respect to tools qualification aspects. [AD19].

4.2 ML Data Management Tools

Data management tools include dataset collection, generation, preparation and verification tools. Training and validation datasets are essential development artifacts that directly contribute to the quality of ML models and downstream implementation. However, there is no intrinsic dissimilarity between training and testing datasets compared to the classical development life cycle where development and testing artifacts are fundamentally different entities (classical testing artifacts include test inputs and expected results while development artifacts specify the implementation design). This aspect blurs the boundary between development and verification artifacts with respect to training and test datasets and therefore the direct applicability of the tool qualification Criteria 1 to training/validation data tools and Criteria 3 to test data tools as per DO-178C [RT11a] should be revisited.

DO-200B [RT15] is a recognised standard for aeronautical data management and can be proposed as a suitable basis for ML data management tool qualification considerations. However, DO-200B [RT15] significantly reduces the qualification levels for data management tools (Tab. 3, Tab. 4) comparing to the DO-178C [RT11a] mapping between the tool qualification levels and failure condition categories. (Tab. 2).

To address the described above concerns about DO-178C and DO-200B approaches with respect to qualification aspects of data management tools in ML workflows, we propose two different scenarios with for tool qualification level assignment:

- Different tools developed independently are used for training/validation data and test data management. In this case there is no potential common source of errors and less stringent DO-200B approach can be followed: TQL-5 for data verification tools and TQL-3/4 for data processing tools. This scenario with independent datasets requires more development efforts and should be applied for assurance levels A and B to address the DO-178C concept of verification independence.
- Same tool is used for both training/validation data and test data management. In this case such tool can be a source of common error in both train and test data and the more rigorous DO-178C approach should be used, i.e. Criteria 1 for the data processing tools (e.g. synthetic data generation tools). For data verification tools the more stringent Criteria 2 is deemed relevant in order to mitigate the lack of independence between development and verification artifacts that inherently exist in classical software development life cycle. This scenario can be relevant for lower assurance levels C and D that do not require independent verification in traditional DO-178C workflows.

4.3 ML Model Development Tools

ML model building, training, and optimization tools are directly involved in the production of a trained and optimized ML model that establishes the functionality of the target system and therefore are *development tools* in terms of [RT11a, RT11b]. ML training tools are at the core of ML development workflows implementing the complex iterative process of ML model parameter computation using the training datasets.

Current certification standards require verification of several properties of development artifacts such as compliance with upstream requirements, accuracy, consistency, verifiability, compatibility with the target computer, conformance to standards, and traceability. These verification credits can be potentially obtained through the qualification of relevant functionality of ML development tools. If verification credit can be fully achieved through the use of an ML development tool it is considered as a traditional tool qualification scenario per DO-330 [RT11b] and the tool must be classified as Criteria 1 tool. Examples of such credit can include conformance to standards (e.g. required model properties such as data types are preserved during training) and compatibility with the target computer (e.g. compatible data types of model parameters are preserved during training).

However, for ML models not all conventional verification objectives can be achieved via direct verification or development tools qualification. The analysis conducted by EUROCAE/SAE working group WG-114/G-34 [EU21, SA21] shows that verification of traceability between training data and trained ML model of a useful size cannot be practically performed due to the entanglement of the iterative learning process and very high complexity of useful ML models. Traceability analysis is used to achieve two verification goals:

- Demonstrate that all upstream requirements are implemented in design models (traceability from training data to ML model)
- Demonstrate that design models have no unintended functionality which is not traceable to upstream requirements (traceability from ML model to training data).

We propose to use qualification of ML training framework capability for training data processing as a mitigation mean to ensure that all elements of training datasets are correctly handled by the training algorithm and reflected in the trained model. This corresponds to the first goal of the conventional traceability analysis but doesn't address the second goal and therefore cannot provide full credit for the traceability verification objective. Furthermore, the overall excessive complexity of ML models weakens the concept of traceability in general so it needs further mitigation using other independent verification means. Therefore, the highest qualification Criteria 1 is an excessive for the partial traceability credit of ML training tools and we propose to apply Criteria 2 as a balanced approach.

Similarly, Criteria 2 can be applied to other capabilities of ML training tools that are positioned as mitigation measures for the know ML verification gaps. For example, out-of-distribution detectors embedded in ML models during training that can serve as an ancillary mean for mitigation of ML unintended behaviour.

4.4 Custom Criteria Assignment for ML-specific Tools

The considered above approach for addressing the distinct properties of ML-specific tools can be formalized as the following expansions to the definitions of DO-178C tool impact criteria:

- Extension of the DO-178C Criteria 2 to ML data verification tools if the same verification tool is applied for both training and testing datasets.

- New criteria for ML data development and verification tools based on the DO-200B levels mapping (Tab. 4) if independence of tools for training and testing datasets is ensured.
- Extension of the DO-178C Criteria 2 to the capabilities of ML training tools that can serve as additional mitigation measures for increasing ML model verification confidence.

5 Conclusion and Future Work

The integration of ML technology in airborne systems extends the currently established practices. The community is trying to build required level of trust to get systems airworthy using cascaded verification strategies that renders all the newly introduced life-cycle process activities for machine learning. There are still verification gaps yet to be covered.

In this paper we review the types of tools specific to ML workflows with respect to the tool qualification level assignment and propose to elaborate the DO-178C impact criteria for specific cases of ML data management and ML model training tools.

This paper also investigates the possibilities where tool qualification can be further exploited. That leads to a new specification of tool qualification criteria for ML-specific tools. Thereby, the tools are classified based on their operations, which determine the amount of rigor for their qualification. But, the paper contributes with starting a further discussion about how partial certification credit can be collected through tool qualification to help covering the above-mentioned gaps. That is rather a novel approach, where tool qualification contributes to the overall airworthiness of the ML-based system, even if the outputs of the tools are still verified.

The follow up of this paper is a work in progress that is exercising the approach explained in this paper for application use cases. The concrete examples are supposed to develop concrete evidence for validity, and feasibility of the proposed approach.

Bibliography

[AD19] Aravantinos, Vincent; Diehl, Frederik: Traceability of Deep Neural Networks, 2019.

[DSH21] Dmitriev, Konstantin; Schumann, Johann; Holzapfel, Florian: Toward Certification of Machine-Learning Systems for Low Criticality Airborne Applications. In: 2021 AIAA/IEEE 40th Digital Avionics Systems Conference (DASC). IEEE, pp. 1–10, 2021.

[DSH22a] Dmitriev, Konstantin; Schumann, Johann; Holzapfel, Florian: Toward Design Assurance of Machine-Learning Airborne Systems. In: AIAA SCITECH 2022 Forum. p. 1134, 2022.

[DSH22b] Dmitriev, Konstantin; Schumann, Johann; Holzapfel, Florian: Towards Design Assurance Level C for Machine-Learning Airborne Applications. In: 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC). pp. 1–6, 2022.

[EA] EASA: AMC20-115D Airborne Software Development Assurance Using EUROCAE ED-12 and RTCA DO-178.

[EA20a] EASA: Artificial Intelligence Roadmap. A human-centric approach to AI in aviation. Technical report, 2020.

[EA20b] EASA: Concepts of Design Assurance for Neural Networks (CoDANN), 2020.

[EA21a] EASA: EASA Concept Paper: First usable guidance for Level 1 machine learning applications. Technical report, 2021.

[EA21b] EASA: Report. Concepts of Design Assurance for Neural Networks (CoDANN) II. Technical report, 2021.

[EU21] EUROCAE: Artificial Intelligence in Aeronautical Systems. Statement of Concerns. Technical Report ER-022, 2021.

[FA22] FAA: AC 20-115D - Airborne Software Development Assurance using Eurocae ED-12( ) and RTCA DO-178( ), 2022.

[Hu20] Huang, Xiaowei; Kroening, Daniel; Ruan, Wenjie; Sharp, James; Sun, Youcheng; Thamo, Emese; Wu, Min; Yi, Xinping: A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review, 37:100270, 2020.

[ID21] Ibrahim, Mohamad; Durak, Umut: State of the Art in Software Tool Qualification with DO-330: A Survey. In (Götz, Sebastian; Linsbauer, Lukas; Schaefer, Ina; Wortmann, Andreas, eds): Proceedings of the Software Engineering 2021 Satellite Events, Braunschweig/Virtual, Germany, February 22 - 26, 2021. volume 2814 of CEUR Workshop Proceedings. CEUR-WS.org, 2021.

[LBH15] LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey: Deep learning. nature, 521(7553):436–444, 2015.

[Ma21] Mamalet, Franck; Jenn, Eric; Flandin, Gregory; Delseny, Hervé; Gabreau, Christophe; Gauffriau, Adrien; Beaudouin, Bernard; Ponsolle, Ludovic; Alecu, Lucian; Bonnin, Hugues et al.: White paper machine learning in certified systems. PhD thesis, IRT Saint Exupéry; ANITI, 2021.

[RT11a] RTCA, Inc.: DO-178C Software Considerations in Airborne Systems and Equipment Certification, 2011.

[RT11b] RTCA, Inc.: DO-330 Software Tool Qualification Consideration. Washington DC, 2011.

[RT11c] RTCA, Inc.: DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A. Washington DC, 2011.

[RT15] RTCA, Inc.: DO-200B Standards for Processing Aeronautical Data. 2015.

[SA21] SAE: Artificial Intelligence in Aeronautical Systems. Statement of Concerns. Technical Report AIR6988, 2021.

[Ta22] Tambon, Florian; Laberge, Gabriel; An, Le; Nikanjam, Amin; Mindom, Paulina Stevia Nouwou; Pequignot, Yann; Khomh, Foutse; Antoniol, Giulio; Merlo, Ettore; Laviolette, François: How to certify machine learning based safety-critical systems? A systematic literature review. Automated Software Engineering, 29(2):38, 2022.

[TDD22] Torens, Christoph; Durak, Umut; Dauer, Johann C.: Guidelines and Regulatory Framework for Machine Learning in Aviation. In: AIAA SCITECH 2022 Forum. January 2022.

[To23] Torens, Christoph; Juenger, Franz; Schirmer, Sebastian; Schopferer, Simon; Zhukov, Dmytro; Dauer, Johann C: Ensuring Safety of Machine Learning Components Using Operational Design Domain. In: AIAA SCITECH 2023 Forum. p. 1124, 2023.

[Vi21] Vidot, Guillaume; Gabreau, Christophe; Ober, Ileana; Ober, Iulian: Certification of embedded systems based on Machine Learning: A survey. arXiv preprint arXiv:2106.07221, 2021.