Safety Assessment of a Machine Learning-Based Aircraft Emergency Braking System: A Case Study

Konstantin Dmitriev*, Julian Rhein†, Lukas Beller‡, Johannes Bröcker§, Evangelos Huber¶,
Johann Schumann‖, Florian Holzapfel**
*†‡§¶** *Institute of Flight System Dynamics*, *Technische Universität München*, Munich, Germany
*konstantin.dmitriev@tum.de, †julian.rhein@tum.de, ‡lukas.beller@tum.de,
§johannes.bröcker@tum.de, ¶evangelos.huber@tum.de, **florian.holzapfel@tum.de
‖ *KBR, NASA Ames Research Center*, Moffett Field, CA
‖ johann.m.schumann@nasa.gov

Abstract—**Machine Learning (ML) is revolutionizing many technological fields, but its use in aviation remains restricted due to stringent certification requirements. Efforts by the aviation community to establish standards for certifying ML-based systems are progressing, yet challenges persist, particularly with safety assessment methods for ML-based systems. This research addresses these challenges through a case study of an autonomous emergency braking system utilizing a computer vision deep neural network (DNN). We demonstrate a safety assessment process tailored to ML-specific concerns, such as low integrity and performance variability in quantitative safety analysis. This study can serve as an illustrative example to facilitate the discussion and convergence on certification aspects for ML-based systems within the aviation community.**

I. Introduction

Machine Learning is rapidly transforming information and transportation technologies; however, it has not been utilized in aviation systems due to incompatibilities with conservative certification practices for safety-critical airborne applications. In recent years, aviation authorities, standardization bodies, and industry stakeholders have collaborated to establish a regulatory framework for certifying ML applications in airborne systems. According to the roadmap recently published by the European Union Aviation Safety Agency (EASA) [1], these efforts are now in the final stages of formulating harmonized means of compliance for the certification of ML-based aircraft systems.

However, some of the ML certification issues are not yet addressed, particularly the methods to manage the performance limitations common to ML models in the context of quantitative safety assessment of aircraft systems. This topic is currently under discussion in the joint EUROCAE/SAE WG-114[1]/G-34[2] working group. Existing standards, such as ARP4761A/ED-135 [2], [3] and ARP4754B/ED-79B [4], [5], do not provide explicit guidance for considering performance aspects in quantitative safety analysis.

In this work, we aimed to study and address the impact of ML-specific aspects on the safety assessment process of aircraft systems. To illustrate the problem, we conducted a case study on the safety assessment of an Aircraft Emergency Braking System (AEBS) that utilizes a deep neural network (ML component) for the visual detection of airport signs. In particular, we demonstrated how the performance-related safety requirements for a state-of-the-art computer vision DNN can be established and verified. Our case study specifically addresses ML-related issues such as the low integrity of image detection DNNs, which leads to spurious and missed detections, as well as variability in performance.

The overall study introduces novel methodologies for addressing ML-specific challenges in safety assessments and aims to serve as a reference example to facilitate the aviation community's convergence on certification standards for ML-based systems. By providing practical insights, this study promotes the adoption of advanced ML technologies in aviation and contributes to the overall safety and efficiency of future airborne applications.

The rest of this paper is structured as follows: Section II reviews relevant literature from industry, regulatory bodies, and academia, presenting the current state of the art on the topic. Section III provides an overview of the conventional process for safety assessment of aircraft systems and introduces a custom approach intended to address the performance limitations inherent in ML-based systems. In Section IV we present a case study of an aircraft system that demonstrates the practical application of the safety assessment process to an ML-based system. Finally, Section V consolidates the key outcomes of the study and suggests future research work.

II. Related Work

The key standardization activities in the field of Machine Learning and Artificial Intelligence (AI) certification in the aviation domain are conducted within the EUROCAE/SAE WG-114/G-34 joint working group, which has been developing the industry standard ARP6983 for ML/AI certification since 2019. The working group initially published the Statement of Concerns [6], followed by several publications by group members discussing various aspects of ML assurance to be addressed in the forthcoming standard [7]–[13].

The European Union Aviation Safety Agency (EASA) and the U.S. Federal Aviation Administration (FAA) are actively involved in the efforts of the WG-114/G-34 working group.

[1] https://www.eurocae.net/about-us/working-groups/

[2] http://profiles.sae.org/teag34/

Additionally, EASA and FAA have issued publications outlining their vision for concepts of ML assurance [14]–[16]. In particular, the EASA concept paper [15] includes anticipated objectives and means of compliance for the safety assessment of ML applications. These objectives and means of compliance are currently under discussion within the WG-114/G-34 working group to harmonize the various approaches envisioned by the group members. The EASA CoDANN reports [17], [18] and the FAA report [16], prepared in partnership with Daedalean AG, include examples of safety assessment activities for a DNN-based vision system. However, only a subset of these materials has been released in public reports. For example, the key assumption about the impact of autocorrelation of DNN outputs on safety requirements verification is not detailed.

The detailed guidance for the safety assessment process of conventional aircraft systems is available in the well-established standards ARP4761A/ED-135 [2], [3], ARP4754B/ED-79B [4], [5], EASA Certification Specification CS-25 [19], and FAA advisory circulars [20], [21]. While the methods and processes described in these existing standards are relevant to ML-based systems, they do not explicitly address the performance limitations inherent to ML technology that we aim to study as a key part of this work.

III. Safety assessment aspects of ML-based systems

A. Safety Assessment of Aircraft Systems

The overall goal of aircraft and system safety assessment is to demonstrate an adequate level of safety by showing that the risk of potential failure conditions is acceptable. This is based on a systematic evaluation of all aircraft and system functions and design with respect to the overall safety objectives defined by certification regulations (e.g., EASA CS-25.1309 [19]). A harmonized methodology for conducting the safety assessment process has been published in the guideline ARP4761A/ED-135 [2], [3].

This safety assessment process starts with the assessment of hazards resulting from malfunctions using a aircraft/system functional hazard assessment (FHA). An example is given later in Section IV-C1. Based on the classification of hazards, safety objectives are derived for the respective functions.

Those objectives are broken down into system and item safety requirements during the preliminary aircraft/system safety assessment processes (PASA/PSSA). In this step, normally fault tree analysis (FTA) or an equivalent method is used to show how subsystem/equipment failures of a draft system architecture contribute to the analyzed failure conditions and to budget failure probabilities to the individual failure events. The application to our case study is shown in Section IV-C2.

Later during the development as the design matures, an aircraft/system safety assessment (ASA/SSA) is performed, which verifies compliance with the safety objectives for the actual system. This involves providing evidence for both qualitative and quantitative safety requirements. The latter part is largely based on hardware reliability estimation techniques (e.g., statistical analysis of in-service data, prediction using handbook methods, etc.). The extension of this process to include verification of quantitative safety requirements of the ML-based runway sign classifier is described in Section IV-H.

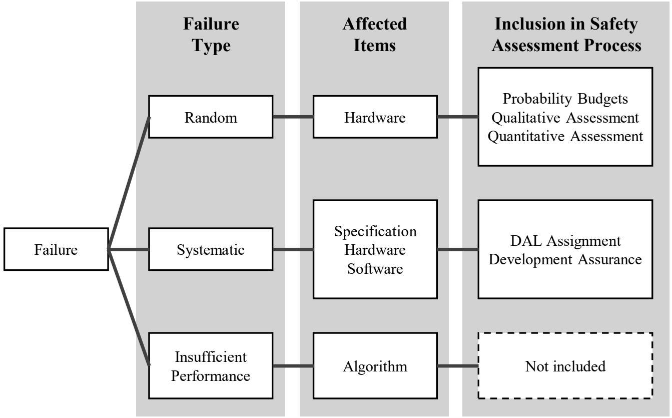

Fig. 1: Failure types and coverage by aerospace safety assessment process

Due to the complexity of aircraft systems the risk of errors in the development of functions and the respective software and electronic hardware items is recognized. Development errors might lead to unintended behavior that can contribute to failure conditions and thus affect safety. The possibility of development errors is addressed by a structured development assurance process which defines validation and verification activities to be performed in order to increase the confidence that a design fulfills its intended function (for details refer to [4], [22], [23]). The level of rigor for those activities is defined by the Development Assurance Level (DAL) that is assigned based on the criticality of the function during the PASA/PSSA phases of the safety assessment process (see [2, Section 3.9]). The DAL assignment is usually based on a qualitative assessment that analyses how development errors of different functions and/or hardware/software items can contribute to a failure condition. While this analysis also uses FTA or a similar technique, in the case of DAL assignment the contributing events represent development errors rather than physical equipment failures and the analysis remains purely qualitative. The DAL assignment for the example application is described in Section IV-C2.

B. Consideration of ML Aspects in Safety Assessment

The probabilistic assessment performed as part of the PASA/PSSA and ASA/SSA phases is normally limited to random hardware failures. It does not cover the possibility of software development errors or other systematic failures because, as stated in DO-178C, ”software reliability rates based on software levels cannot be used by the system safety assessment process” due to limitations of the know software reliability models. It is assumed that software functions developed in accordance with DO-178C [22] are free of known errors to the required qualitative level of confidence. That assumption can not be made for ML-based functions: we assume that an ML-based function will normally be subject to limited performance, in the sense that under normal (failure-free) operating conditions, there is a residual probability that the output of the ML function significantly deviates from the expected output. In the application presented in Section IV, this is given by the residual false-positive and false-negative rates of the ML-based sign detector. As summarized in Fig. 1, the current framework of ARP4761A/ED-135 [2], [3] does not include guidance for consideration of such known performance deficiencies in the safety assessment process: random hardware failures are taken into account by a) assigning a probability budget b) deriving additional qualitative objectives and c) demonstrating that the design meets the probability targets and qualitative objectives during the ASA/SSA phases. Systematic errors that might affect the specification of hardware or software items are taken into account by a) assigning a required development assurance level during the PASA/PSSA phases and b) fulfilling the required development assurance objectives during the system implementation. Considering that the contribution of inherent performance limitations of ML-based functions to the overall risk might be equally significant as random hardware failures or systematic errors, in the following section, a concept for inclusion of this aspect into the safety assessment process is presented.

It should be noted that other functions exist that are subject to the described type of performance limitations. Two examples are autoland (where uncertain performance variation has to be assumed due to varying external conditions such as wind, turbulence, other atmospheric conditions, runway conditions, etc.) and satellite navigation (where performance can not be completely predicted due to external factors such as satellite clock accuracy, ionospheric disturbances, reflections, etc.). However, in the existing certification framework, those functions are usually covered by dedicated performance standards such as [24] in the case of autoland, and [25], [26] for satellite navigation performance. Developing a dedicated performance standard for ML-based functions is not a feasible approach, because it can be expected that ML-based algorithms could be employed for various functions of different criticality and performance requirements in the future that would necessitate dedicated safety assessment for each use-case.

IV. Case Study

In this case study, we demonstrate the safety assessment process and the relevant development activities of the Aircraft Emergency Braking System (AEBS), building upon our previous work on the ML-based Runway Sign Classifier (RSC) [27], [28]. The sign detection capability of the RSC is leveraged to implement a specific safety-critical function within the AEBS.

The AEBS is designed as a pilot assistance system intended to prevent aircraft from entering airport restricted areas marked by ”No Entry” signs (Fig. 5). The specification for such signs is standardized in the FAA Advisory Circular 150/5340-18G [29]. The AEBS actively monitors and identifies ”No Entry” signs during taxi operations and estimates the aircraft's position in relation to these detected signs. If a sign is detected at a distance that may result in entering restricted areas, the system initiates a pilot warning annunciation and subsequently activates emergency braking. The AEBS is intended to serve as an additional safety layer, particularly in the event of crew distraction or incapacitation. Therefore, the flight crew is instructed to consistently perform manual monitoring for the presence of signs during all taxi operations, irrespective of the AEBS's availability.

A. AEBS Development Life Cycle Overview

The development of the AEBS follows the principles outlined in ARP4754B/ED-79B [4], [5] and ARP4761A/ED-135 [2], [3]. This case study primarily focuses on the system-level development and safety assessment activities impacted by the introduction of ML technology, particularly the use of real-world data for safety requirements verification, as detailed in the following subsections.

The development of aircraft-level functions and architecture is a technology-agnostic activity and is not impacted by ML-specific aspects; therefore, it is not considered in this work. Software and hardware development aspects are described in our previous works [27], [28] and are not further discussed here. We reuse and reference the relevant ML development and verification artifacts from our prior work on the Runway Sign Classifier [27], [28] as necessary.

B. AEBS Functions and Architecture

Table I contains three system functions that define the AEBS concept. This paper further illustrates the safety assessment process exemplary for the functions *AEBS-F1* and *AEBS-F2*, as they directly depend on the ML-based computer vision capabilities of the AEBS.

TABLE I: AEBS Functions

ID	Function definition
AEBS-F1	Alert the flight crew when the aircraft approaches the restricted airport area marked by ”No Entry” signs at a distance sufficient for the pilot to react.
AEBS-F2	Automatically initiate emergency braking if the aircraft might enter the restricted airport area marked by ”No Entry” signs unless immediate braking is applied.
AEBS-F3	Monitor system health and inform the flight crew about system failures.

To implement the defined system functions, we proposed a system architecture depicted in Fig. 2. We reused the components from the original RSC system introduced in [27], which implements the perception part of the AEBS, including the video camera and the RSC DNN for ML-based sign detector (MLD).

The Emergency Braking Controller (EBC) was introduced into the AEBS to implement decision logic for proximity alerts and braking activation functions. The EBC also provides interfaces with the Flight Warning System (proximity alert signal for *AEBS-F1*), the Aircraft Braking System (emergency

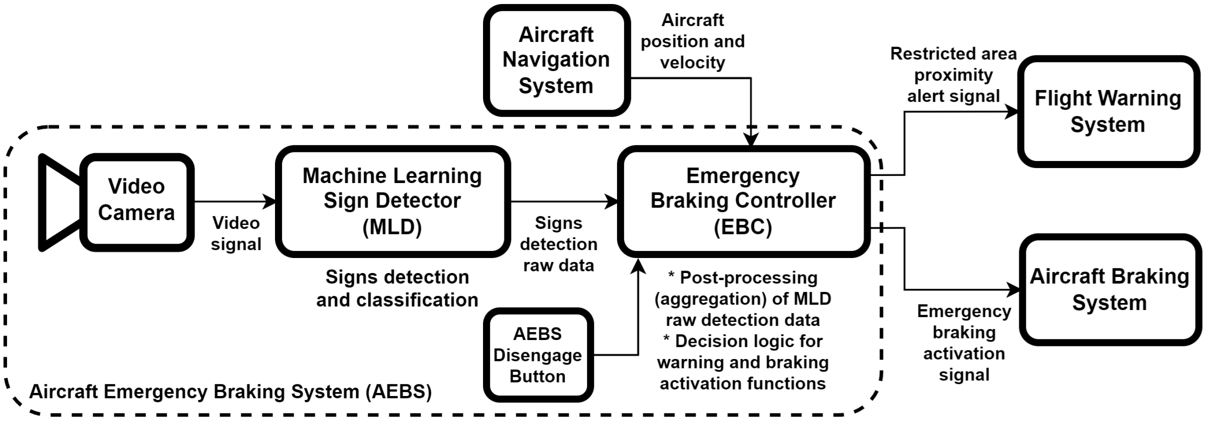
Aircraft Navigation System
Aircraft position and velocity
Restricted area proximity alert signal
Flight Warning System
Video Camera
Video signal
Machine Learning Sign Detector (MLD)
Signs detection and classification
Signs detection raw data
Emergency Braking Controller (EBC)
* Post-processing (aggregation) of MLD raw detection data
* Decision logic for warning and braking activation functions
AEBS Disengage Button
Aircraft Braking System
Emergency braking activation signal
Aircraft Emergency Braking System (AEBS)

Fig. 2: AEBS architecture.

braking activation signal for *AEBS-F2*), and the Aircraft Navigation System (obtaining aircraft position and speed used for emergency braking decision logic *AEBS-F2*).

To ensure the integrity of the EBC decision logic given the expected performance variations of the MLD, the EBC also implements post-processing of the MLD raw detection data. This is achieved by estimating the distance to the sign by averaging several detection samples. The distance is calculated from the size of the sign's image, to which it is inversely proportional due to the fixed focal length of the video camera. If the estimated distance is below the determined warning threshold, the EBC generates a signal to the Aircraft Flight Warning System to inform the flight crew about the proximity of the ”No Entry” sign. If the estimated distance is below the threshold at which immediate braking must be applied to avoid entering the restricted area, the EBC generates a signal to initiate emergency braking.

The number of samples to be averaged during post-processing, the distance thresholds, the frequency of the MLD, and other relevant parameters are not predetermined. Instead, they are iteratively developed throughout the requirements engineering activities and validated in the preliminary safety assessment, as detailed in subsequent sections.

C. AEBS Safety Assessment

In this case study, we conduct the safety assessment process of AEBS following the ARP4761A/ED-135 [2], [3] methodology. Some aspects of the Fault Tree Analysis were elaborated with additional considerations to address performance limitations common to ML-based systems, as proposed in Section III. We also studied in detail the definition of ML performance requirements from the probability budget allocated for ML failure modes related to insufficient performance.

We limited the safety assessment of AEBS to system level, since aircraft-level functions are technology-agnostic and not affected by the ML-specific issues we addressed in this work. The conducted AEBS safety assessment activities are detailed below.

1) Functional Hazard Analysis: The inputs to the system FHA are the AEBS functions (Table I), for which we identified possible failure conditions. Following industry practices, we considered total loss (TL) and malfunctions (MF) for each function and took into account crew awareness (using suffixes ”A” for aware and ”U” for unaware in the failure condition IDs). The identified failure conditions are listed in Table II.

TABLE II: AEBS Failure Conditions

ID	Failure Condition
AEBS-F1.TL.A	Loss of restricted areas proximity alert with crew aware.
AEBS-F1.TL.U	Loss of restricted areas proximity alert with crew unaware.
AEBS-F1.MF	Inadvertent restricted area proximity alert.
AEBS-F2.TL.A	Loss of the automatic emergency braking command with the crew aware.
AEBS-F2.TL.U	Loss of the automatic emergency braking command with the crew unaware.
AEBS-F2.MF	Inadvertent activation of the automatic emergency braking.
AEBS-F2.MF.E	Early activation of the automatic emergency braking.
AEBS-F2.MF.L	Late activation of the automatic emergency braking.

Subsequently, we analyzed the severity of the failure conditions by determining their impact on the aircraft, flight crew, and other occupants. Each identified failure effect is classified in accordance with the hazard classification provided in [20]. The following assumptions have been taken into account while performing this analysis:

- The analysis was conducted for the flight phase ”Taxi” only, as the AEBS is disengaged during other phases.
- The flight crew is instructed to perform consistent manual monitoring for the presence of signs during all taxi operations, irrespective of the AEBS's availability.
- Partial function losses are not considered due to the discrete nature of the function outputs (both proximity alert signal and auto brake command can only have TRUE/FALSE states).

The determined failure conditions, their effect, and respective classifications are summarized in Table III. The established

TABLE III: AEBS Failure Effects and Classification

ID	Failure Effect	Classification
AEBS-F1.TL.A	**Aircraft:** Reduced situation awareness, resulting in a slight reduction of safety margins. **Flight crew:** The crew continues to monitor the presence of signs manually; no significant crew workload impact. **Other occupants:** No direct effect.	Minor
AEBS-F1.TL.U	**Aircraft:** Reduced situation awareness, resulting in a slight reduction of safety margins. **Flight crew:** Slight increase in workload to recognize the failure in the presence of signs and disengage the system. The crew continues to monitor the presence of signs manually. **Other occupants:** No direct effect.	Minor
AEBS-F1.MF	**Aircraft:** Reduced situation awareness, resulting in a slight reduction of safety margins. **Flight crew:** Slight increase in workload to recognize the failure and disengage the system. The crew continues to monitor the presence of signs manually. **Other occupants:** No direct effect.	Minor
AEBS-F2.TL.A	**Aircraft:** Reduced automation, resulting in a slight reduction of safety margins. **Flight crew:** The crew continues to monitor the presence of signs manually and take necessary control actions. **Other occupants:** No direct effect.	Minor
AEBS-F2.TL.U	**Aircraft:** Reduced safety features, resulting in a slight reduction of safety margins. **Flight crew:** Slight increase in workload to recognize the failure. The crew continues to monitor the presence of signs manually and take necessary control actions. **Other occupants:** No direct effect.	Minor
AEBS-F2.MF	**Aircraft:** No direct effect. **Flight crew:** The crew will observe the condition and abort the taxi operation. Potential injury to unrestrained cabin crew due to sharp deceleration. **Other occupants:** Potential injury to unrestrained occupants due to sharp deceleration.	Major
AEBS-F2.MF.E	The failure effect is the same as for AEBS-F2.MF	Major
AEBS-F2.MF.L	The failure effect is the same as for AEBS-F2.TL.U	Minor

classification of these failure conditions sets the functional safety objectives for the AEBS. Due to space constraints, we focus exclusively on the failure conditions for *AEBS-F1* in the subsequent sections. It is important to note that, at this stage, the failure condition AEBS-F1.TL.U is apparently linked to the specific *false negative* failure mode of MLD (FN_{mld}). The FN_{mld} failure mode is defined as follows: a sign is present in the field of view under nominal operating conditions, but the MLD returns an empty detection vector. This association is confirmed and further detailed in the subsequent fault tree analysis.

2) Fault Tree Analysis: We used quantitative FTA to allocate failure probability budgets for the AEBS hardware items and included the events associated with insufficient performance of the MLD. Here, we present the fault tree for one representative failure condition, *AEBS-F1.TL.U*, noting that the same approach can be applied to other failure conditions.

Although FAA AC 25.1309 [20] and EASA AMC 25.1309 [19] do not require FTAs (qualitative or quantitative analysis) for minor failure conditions, we utilize them to illustrate our approach, which can be applied to any failure condition category. Furthermore, given the novelty of ML technology, we argue that it is prudent to budget probabilities even for minor ML failure conditions to establish robust performance requirements.

Figure 3 presents the quantitative fault tree for the selected failure condition *AEBS-F1.TL.U* (see Table III). The basic events associated with hardware random failure are shown in the left frame. The label denoted with FR below the basic events represents the budgeted failure rate. For illustrative purposes, we assume budgets for random hardware failure probabilities based on typical failure rates provided in Appendix Q of ARP4761A/ED-135 [2], [3]. The exposure time is the assumed average flight duration of $T = 4\,h$. Although the AEBS function is active only during taxi phases, the entire flight duration is taken as exposure time, because it is assumed that the required hardware can still fail at any point during the flight.

The distinctive failure event *EBC_FAIL_TRK*, associated with insufficient ML performance, is defined as ”EBC fails to track a sign due to the excessive density of false negative MLD detections FN_{mld}” and is shown in the right frame in Figure 3. A second cause for missing proximity alert is given by the event *EBC_ERR_TRK*, which describes the situation where a proximity alert is not generated because the distance to the sign is overestimated by the EBC.

The probability of the event is modeled by a constant value, which represents the budgeted probability per flight. From this the budgeted probability P_{eft} of failure to track per “No Entry” sign encounter is calculated based on the assumption that on average two “No Entry” signs are encountered per flight:

$$P_{eft} \leq \frac{4.0 \cdot 10^{-4}}{2} = 2.0 \cdot 10^{-4} \qquad (1)$$

This is captured as a system safety requirement, upon which the necessary ML performance characteristics are determined in the next subsection.

The labels denoted with Q in Fig. 3 represent the per-flight probability for the respective gates and events. The cumulative probability of the event *AEBS-F1.TL.U* is approximately $2.4 \cdot 10^{-3}$ per flight, which corresponds to $6.0 \cdot 10^{-4}$ per flight hour. This value complies with the requirement of a probability lower than $1.0 \cdot 10^{-3}$ per flight hour for minor failure conditions and allows some additional budget for the undeveloped random hardware failure events.

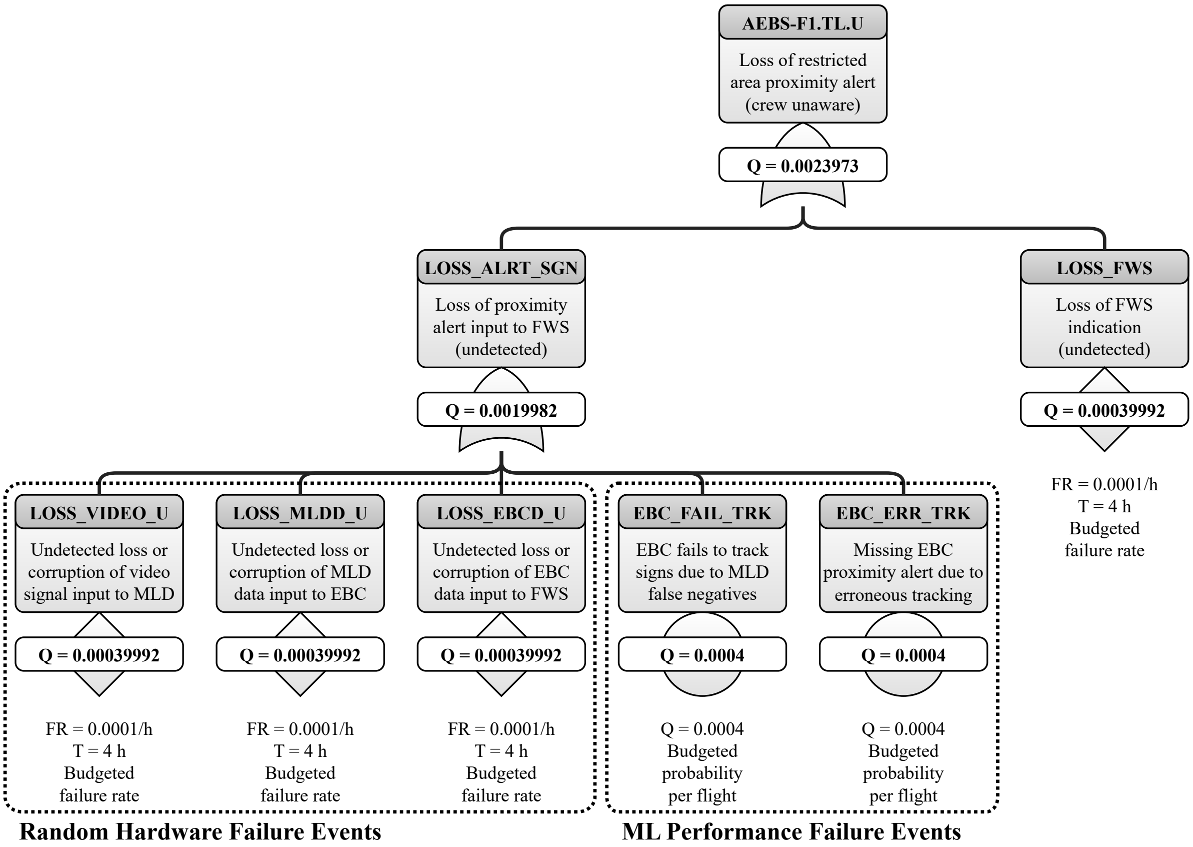
AEBS-F1.TL.U
Loss of restricted area proximity alert (crew unaware)
Q = 0.0023973
LOSS_ALRT_SGN
Loss of proximity alert input to FWS (undetected)
Q = 0.0019982
LOSS_FWS
Loss of FWS indication (undetected)
Q = 0.00039992
FR = 0.0001/h
T = 4 h
Budgeted failure rate
LOSS_VIDEO_U
Undetected loss or corruption of video signal input to MLD
Q = 0.00039992
FR = 0.0001/h
T = 4 h
Budgeted failure rate
LOSS_MLDD_U
Undetected loss or corruption of MLD data input to EBC
Q = 0.00039992
FR = 0.0001/h
T = 4 h
Budgeted failure rate
LOSS_EBCD_U
Undetected loss or corruption of EBC data input to FWS
Q = 0.00039992
FR = 0.0001/h
T = 4 h
Budgeted failure rate
EBC_FAIL_TRK
EBC fails to track signs due to MLD false negatives
Q = 0.0004
Q = 0.0004
Budgeted probability per flight
EBC_ERR_TRK
Missing EBC proximity alert due to erroneous tracking
Q = 0.0004
Q = 0.0004
Budgeted probability per flight
Random Hardware Failure Events
ML Performance Failure Events

Fig. 3: Fault tree for the failure condition AEBS-F1.TL.U

The assignment of DALs to the hardware and software items of the architecture is straightforward, based on the fact that there is no (function or item) development independence in the architecture. That means that the item DAL (IDAL) for all items is selected as the highest among all function DALs (i.e. case 1 according to [3, Section P.7]). Given that a malfunction of F2 can result in a major effect, the DAL of all items has to be selected as C or higher.

3) Definition of Safety Requirements: The determined hardware failure probability budgets and IDALs directly translate to safety requirements for the AEBS components. However, the probability budget allocated to the *EBC_FAIL_TRK* event, associated with insufficient ML performance, requires further engineering refinement because several characteristics of AEBS components can impact the event probability. We conducted a detailed analysis of the system function associated with this failure event to identify and capture all relevant safety requirements for system components as follows. This serves as an illustration of the generic approach for refining safety requirements related to ML performance.

The following assumptions about the characteristics of the aircraft are provided as inputs to the analysis of the failure condition *AEBS-F1.TL.U*:

- Aircraft maximum taxi speed: $V_0 = 30kn \approx 15,43m/s$
- Emergency braking deceleration rate: $a = 6m/s^2$
- Pilot's reaction time (as per AMC 25.671 [19]): $t_r = 3s$

All assumptions correspond to the worst-case scenario, including maximum speed and lowest deceleration rate. This results in the minimum distance at which the sign must be detected and a warning annunciation must be generated:

$$D_{min} = V_0 \cdot t_r + V_0^2/2a \approx 66m \tag{2}$$

The ML detector characteristics established in previous works [27], [28] are used as assumptions to determine the preliminary characteristics of the *detection window* within which the system must reliably detect a ”No Entry” sign. The braking scenario corresponding to the AEBS-F1.TL.U failure condition and the associated *detection window* are illustrated in Figure 4. Given the assumptions based on the evaluation of the RSC characteristics in previous work [28] (maximum detection distance $D_{max} = 85m$ and the DNN detection frequency f_{det} is at least 10 Hz) the detection window length D_{det} and the minimum number of detection samples within the *detection window* n can be computed:

$$D_{det} = D_{max} - D_{min} \approx 19m \tag{3}$$

$$n = \lfloor D_{det}/(V_0) \cdot f_{det} \rfloor = 12. \tag{4}$$

We selected the minimum number of positive detections samples within the *detection window* to establish the sign track

$n_p = 5$. This selection aims to achieve the target probability of *EBC_FAIL_TRK* event, and yields the minimum number k_0 of FN_{mld} events that would result in failure to track a sign (*EBC_FAIL_TRK* event)

$$k_0 = n - n_p + 1; \tag{5}$$

The relationship between the probabilities P_{eft} of the *EBC_FAIL_TRK* event and p of the FN_{mld} event can be approximated using the Poisson distribution, assuming the FN_{mld} events are independent, infrequent and equally probable:

$$P_{eft} \approx \sum_{k=k_0}^{n} \frac{e^{-\lambda}\lambda^k}{k!} \text{where } \lambda = p \cdot n \tag{6}$$

While binomial distributions can relate the p and FN_{mld} probabilities more precisely, the Poisson distribution was selected for its illustrative simplicity. The equation 6 is solved numerically to find the probability p of the FN_{mld} event that corresponds to the target probability $P_{eft} \leq 2 \times 10^{-4}$ for the event *EBC_FAIL_TRK*. The solution for the worst-case scenario (maximum speed, which corresponds to the smallest number of samples n in the detection window) determines that $p \lesssim 0.13$; for larger n, p increases. This value and other parameters used in this analysis are captured as safety requirements in Subsection IV-D.

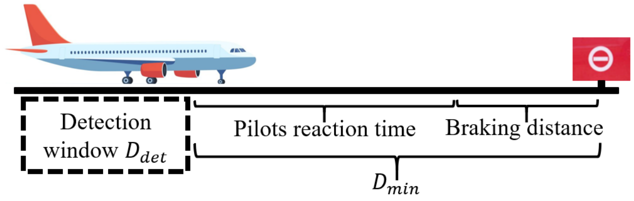

Fig. 4: Braking scenario for the AEBS F1.TL.U failure condition

D. Requirements Capture and Validation

Functional and operational requirements for AEBS are developed from the system functions and operational assumptions. Safety requirements are captured based on the safety analysis (see Subsection IV-C3). We also integrate the applicable requirements for the RSC system [27], [28] which is used as a basis for this development. A subset of the requirements for the AEBS function AEBS-F1 is presented in Table IV. The following definitions are utilized in the requirements:

- The *EBC_FAIL_TRK* event is defined as follows: a sign is present in the field of view, but the EBC fails to establish a track of the sign due to an excessive rate of FN_{mld} events.
- The detection window for AEBS-F1.TL.U is defined as the distance range within which the system must reliably detect a ”No Entry” sign to generate a timely alert.

System requirements were allocated to the system items and subsequently refined into data and component requirements, consistent with the methodologies proposed and employed in our previous case studies [27], [28]. The validation of all levels of requirements was conducted through a combination of engineering reviews, modeling, and safety assessments.

TABLE IV: AEBS Requirements Subset for AEBS-F1 where SRF# are functional requirements, SRO# are operational requirements and SRS# are safety requirements.

ID	Description
AEBS-SRF1	The AEBS shall generate an alert signal provided to the Flight Warning System to inform the flight crew about the proximity of ”No Entry” signs (AC 150/5340-18G) from a distance that is sufficient to prevent the aircraft from entering the restricted area.
AEBS-SRF2	The AEBS shall operate in visibility conditions corresponding to runway visual range 400 m or above as per ICAO Doc 9328.
AEBS-SRO2	The AEBS shall provide an interface for the flight crew to disengage the proximity alert signal in case of system malfunction.
AEBS-SRS1	Loss of AEBS proximity alert shall be less probable than 1.0E-03 per flight hour.
AEBS-SRS2	The probability of the *EBC_FAIL_TRK* event shall not exceed 2.0E-04 within the detection window associated with the AEBS-F1.TL.U failure condition.
AEBS-SRS3	The probability of the FN_{mld} event shall not exceed 0.13 within the detection window associated with the AEBS-F1.TL.U failure condition.
AEBS-SRS4	The MLD detection frequency shall be at least 10Hz.
AEBS-SRS5	The EBC shall establish a track of the signs when at least 5 detection vectors are received from MLD within the established detection window.

E. ML Data Management Process

For AEBS, we use the ML data management process for synthetic data proposed and implemented for the RSC case study [28] as a foundation. To validate the assumptions made in safety assessment process in realistic operational context, we introduced in this work an extensive real-world dataset described in Subsection IV-F.

For synthetic part of the AEBS dataset, we used the images of airport signs generated for RSC as a baseline, augmenting it with ”No Entry” sign images since these were not sufficiently represented in the original RSC dataset. To generate synthetic images, we utilized a data generation framework based on MATLAB/Simulink[3] and X-Plane[4]/FlightGear[5] flight simulators as described in [27]. The final synthetic dataset, which delivers sufficient detection performance of the trained DNN on both synthetic and real-world data, includes 667 images of ”No Entry” signs in different airports and environmental conditions. This dataset is published in an open-source repository[6].

F. Real World Data Collection

Performing safety analysis, using synthetic data assumes that the synthetic data is similar enough to the real-world

[3] mathworks.com

[4] https://www.x-plane.com/

[5] https://www.flightgear.org/

[6] https://gitlab.com/tum-fsd/runway-sign-classifier-dnn

environment. This assumption is hard to justify, as simulating a visual airport environment is a complex task. Therefore, we introduced a real-world dataset that provides the highest level of confidence for verifying safety requirements and assumptions. We recorded the data on an inoperative airfield that represent the system's operational domain. Figure 5 illustrates one of the video frames from the recorded dataset. The video sequences were recorded using a cellphone camera with a focal length equivalent to 26mm (35mm format), a 3840 x 2160 pixels resolution, and a frame rate of 30Hz. The recordings were post-processed using a custom MATLAB script to emulate the characteristics of the AEBS video camera by:

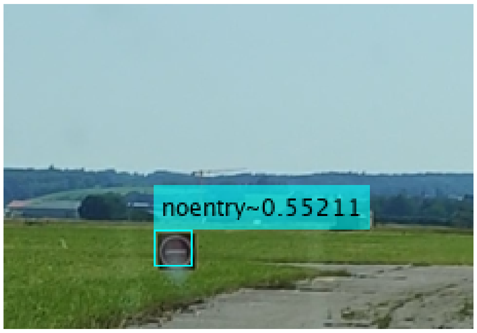

Fig. 5: A real-world dataset sample with No Entry sign and the detection information layer

- Extracting individual video frames as images,
- Cropping the images' area to correspond with the camera's field of view (11°), and
- Rescaling the images to match the input layer size of the MLD DNN (256 x 256 pixels).

By processing recorded videos that capture various sign locations, times of day, and weather conditions to cover the defined data requirements, we produced a real-world dataset comprising 7329 images. The distance between the camera and the sign position shall be used as a ground truth for verification of the EBC tracking algorithm. Thus, we recorded GPS position measurements of the sign and the camera (cam)

$$\vec{x}^{GPS}_{Sign}(t_i) = \begin{bmatrix} \phi_{Sign}(t_i) \\ \lambda_{Sign}(t_i) \end{bmatrix}, \quad \vec{x}^{GPS}_{Cam}(t_i) = \begin{bmatrix} \phi_{Cam}(t_i) \\ \lambda_{Cam}(t_i) \end{bmatrix} \tag{7}$$

where ϕ denotes the latitude, λ denotes the longitude and

$$\Delta t^{GPS} = t^{GPS}_{i+1} - t^{GPS}_{i} = 1s. \tag{8}$$

denotes the recording interval of the GPS receivers. The distance $d(t^{GPS}_i)$ between the sign and the camera at times t^{GPS}_i is approximated using the haversine formula, assuming a spherical earth. Figure 6 illustrates the GPS-based distance approximation results for one video recording session.

G. System Implementation

In this case study, we focus on the implementation of the DNN-based MLD component, which requires specific ML considerations. We apply the custom ML implementation process proposed in [12], [13] and demonstrated for RSC in [27], [28]. Due to space constraints, this paper addresses only the model-level implementation, omitting the deployment of the trained model on an embedded hardware platform. The deployment can be conducted using the approach proposed in [12], [13], based on the conventional ED-12C/DO-178C [22] and ED-80/DO-254 [23] practices.

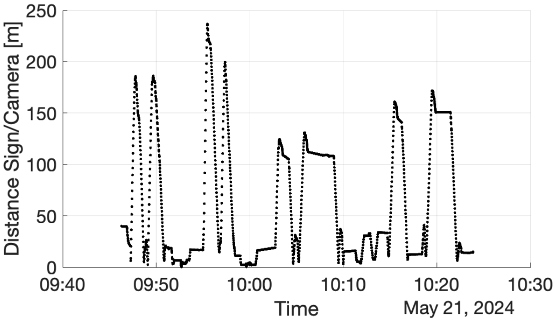

Fig. 6: Resulting distance approximations between camera and sign over recording time, using the haversine formula.

For the MLD implementation, we upgraded the RSC DNN architecture to YOLOv3 [30], which demonstrated better performance than the original YOLOv2 architecture [31] on the dataset augmented with "No Entry" signs. We utilized the MATLAB implementation of the YOLOv3 DNN, `yolov3ObjectDetector`[7] and used the SqueezeNet [32] as a backbone network. The training and tuning were conducted using the MATLAB Computer Vision Toolbox[8]. Given that the MLD component was assigned IDAL C, the implementation of a dissimilar redundant DNN, as proposed in [13] should be undertaken (omitted here for simplicity).

By using the synthetic dataset for YOLOv3 training and tuning, we achieved an average precision of 96.1% for detecting "No Entry" signs and maintained a false negative detection rate (FN_{mld}) below 4% on the synthetic test dataset. This performance meets the safety requirements determined in the preliminary safety assessment (Subsection IV-C3), and consequently, the MLD DNN model was advanced to verification with real-world data.

H. System Verification and Safety Assessment

System verification activities are intended to ensure that the system implementation meets its specified requirements and to detect potential errors introduced during development, as outlined in ARP4754B/ED-79B [4], [5]. With respect to safety objectives and requirements, the System Safety Assessment process ARP4761A/ED-135 [2], [3] is used to verify that established safety requirements are satisfied. In this work, we focus on the verification of ML-specific safety requirements related to the inherent performance limitations of ML-based components. The verification of requirements for conventional non-ML components of the AEBS system, along with the corresponding safety assessment activities,

[7] https://www.mathworks.com/help/vision/ref/yolov3objectdetector.html
[8] https://www.mathworks.com/help/vision/index.html

can be conducted using traditional methods, as detailed in Appendix E of ARP4754B/ED-79B [4], [5] and Appendix Q of ARP4761A/ED-135 [2], [3].

1) Verification of Safety Requirements: To illustrate the verification process for ML-specific performance requirements, we selected the safety requirement *AEBS-SRS3*: "The probability of the FN_{mld} event shall not exceed 0.13 within the detection window associated with the AEBS-F1.TL.U failure condition." The probability of the FN_{mld} event can be estimated by measuring its frequency using a sufficiently representative test dataset, assuming the randomness of the underlying phenomena that result in variations in DNN performance within the nominal range. The primary factor contributing to these performance variations is the high complexity of the DNN model; for example, the YOLOv3 DNN used in this work contains over 6.3 million parameters. The inherent uncertainties of DNN training data and the vast number of resulting DNN parameters lead to random variations in DNN performance, referred to as *aleatoric uncertainty* in [18]. Random variation in DNN performance within the nominal range should be distinguished from the loss of DNN performance outside the nominal range or systematic DNN errors, which are referred to as *epistemic uncertainties* in [18]. Epistemic uncertainties are managed through a learning assurance process [13], [15], which provides confidence that (1) the nominal operating range (ODD) is properly defined and meets the system requirements, and (2) the required DNN generalization capabilities are achieved within the nominal range.

We used the real-world dataset (see Subsection IV-F) to measure the FN_{mld} frequency. The distribution of the MLD DNN detection samples in nominal range is depicted in Figure 7. The relative rate of FN_{mld} events R_{fn} was calculated as

$$R_{fn} = \frac{\text{Number of } FN_{mld} \text{ samples}}{\text{Total number of MLD detections}} \tag{9}$$

The measured R_{fn} was approximately 6.1% within the *detection window* associated with the AEBS-F1.TL.U failure condition (and approximately 44% across the entire dataset). Therefore the requirements *AEBS-SRS3* is satisfied.

2) Confirmation of Assumptions: The assumption made during the system development and safety assessment must be managed and confirmed, because incorrect assumptions "can jeopardize satisfactory implementation of safety requirements" [4, p.44]. During the definition of AEBS safety requirements (Subsection IV-C3), an assumption regarding the independence of FN_{mld} events was made. This assumption is crucial because it used to relate probabilities of FN_{mld} and *EBC_FAIL_TRK*. If this assumption is incorrect, the target probability of FN_{mld} may be determined inaccurately. In the following analysis, we analyze this assumption using autocorrelation analysis of the MLD output time series.

We utilized our real-world dataset to generate time series of MLD detections corresponding to trajectories that represent expected taxiing scenarios. To evaluate the correlation between individual detection samples within the *detection window*, we computed the autocorrelation function (ACF) for the MLD detection time series. These time series were aggregated from the entire dataset. Figure 8 shows the ACF for the aggregated detection time series. We observe a steady decrease of the ACF until lag 10, after which the ACF oscillates around zero.

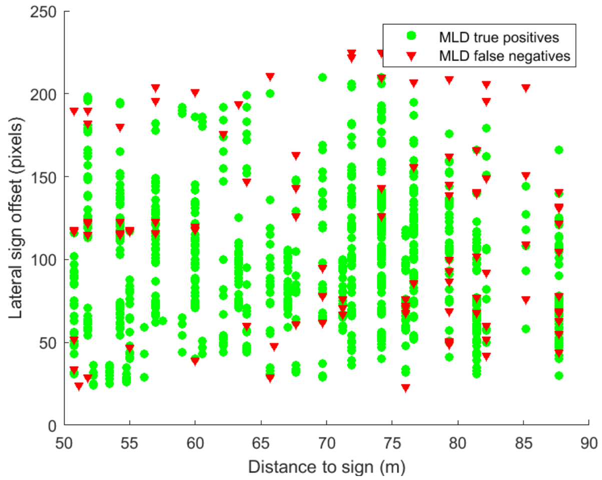

Fig. 7: MLD detection distribution

This indicates that there is no obvious correlation and, therefore, no linear dependency between samples that are more than $N_i = 10$ lags apart. Given the dataset recording frequency $f_{rec} = 30$ Hz and the vehicle speed during recording $V_r \approx 3$ m/s, the distance between uncorrelated samples can be calculated as:

$$d_i = \frac{V_r}{f_{rec}} \cdot N_i \approx 1 \text{ m} \tag{10}$$

This yields the number of uncorrelated samples in the *detection window* for the worst-case scenario with maximum taxiing speed:

$$N = \left\lfloor \frac{D_{det}}{d_i} \right\rfloor = 19 \tag{11}$$

which is greater than the assumed number of uncorrelated samples $n = 12$. Therefore, the assumption used in Equation 6 is at least partially justified; however, the impact of potential higher-order dependencies needs further analysis.

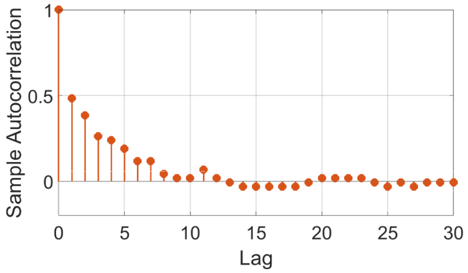

Fig. 8: MLD detection time series autocorrelation function

V. Conclusions and Future Work

In this work, we aimed to address the ML-specific aspects in the safety assessment of aircraft systems. Through a review of the conventional safety assessment process, we demonstrated that the performance limitations inherent in ML-based systems can contribute to system failure conditions and must be accounted for in the quantitative safety analysis.

To illustrate the problem, we conducted a case study of an autonomous emergency braking system that utilizes an ML-based component for the visual detection of airport signs. We performed a safety assessment of a representative ML-based function by following ARP4761A/ED-135 [2], [3] practices, which were tailored to address the performance limitations of the ML-based system component. Specifically, we included the events associated with ML performance loss in the fault tree analysis, enabling us to determine the quantitative safety requirements for ML performance characteristics. Verification of ML performance safety requirements and the confirmation of assumptions were conducted using statistical methods, particularly autocorrelation analysis of the DNN outputs.

The approach demonstrated in this study can serve as a reference for safety assessments of similar systems, promoting the adoption of advanced ML technologies in aviation. For future work, we plan to examine in greater detail the various failure modes of ML-based components to support the Failure Modes and Effect Analysis (FMEA) method and to explore the applications of autocorrelation analysis more deeply.

Acknowledgment

The authors would like to thank the German Bundeswehr/Air Force for providing access to Erding Air Base. Their support enabled us to record the essential real-world dataset within the operational domain of the investigated image recognition software.

The authors also extend their gratitude to the members of EUROCAE WG-114 / SAE G-34 for their valuable feedback and discussions on this case study. Their insights have been instrumental in refining our work.

References

[1] "Artificial intelligence roadmap 2.0. Human-centric approach to AI in aviation," European Aviation Safety Agency, Tech. Rep., 2023.

[2] *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, SAE International Std. SAE ARP4761A, 2023.

[3] *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*, EUROCAE Std. ED-135, 2023.

[4] *Guidelines for Development of Civil Aircraft and Systems*, SAE International Std. SAE ARP4754B, 2023.

[5] *Guidelines for Development of Civil Aircraft and Systems*, EUROCAE Std. ED-79B, 2023.

[6] "Artificial intelligence in aeronautical systems. Statement of Concerns." EUROCAE, Tech. Rep. AIR6988, 2021.

[7] F. Kaakai, K. Dmitriev, S. Adibhatla, E. Baskaya, and et al., "Toward a Machine Learning Development Lifecycle for Product Certification and Approval in Aviation," *SAE Int. J. Aerosp. 15(2):2022*, 2022.

[8] F. Mamalet, E. Jenn, G. Flandin, H. Delseny, C. Gabreau, A. Gauffriau, B. Beaudouin, L. Ponsolle, L. Alecu, H. Bonnin, B. Beltran, D. Duchel, J.-B. Ginestet, A. Hervieu, S. Pasquet, K. Delmas, C. Pagetti, J.-M. Gabriel, C. Chapdelaine, S. Picard, M. Damour, C. Cappi, L. Gardès, F. D. Grancey, B. Lefevre, S. Gerchinovitz, and A. Albore, "White Paper Machine Learning in Certified Systems," Mar. 2021. [Online]. Available: https://hal.science/hal-03176080

[9] F. de Grancey, S. Gerchinovitz, L. Alecu, H. Bonnin, J. Dalmau, K. Delmas, and F. Mamalet, "On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components," May 2024, accepted for publication at ERTS 2024. [Online]. Available: https://hal.science/hal-04575318

[10] C. Gabreau, A. Gauffriau, F. D. Grancey, J.-B. Ginestet, and C. Pagetti, "Toward the certification of safety-related systems using ML techniques: the ACAS-Xu experience," in *11th European Congress on Embedded Real Time Software and Systems (ERTS 2022)*, Toulouse, France, Jun. 2022. [Online]. Available: https://hal.science/hal-03761946

[11] F. Kaakai, S. Adibhatla, G. Pai, and E. Escorihuela, "Data-centric operational design domain characterization for machine learning-based aeronautical products," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2023, pp. 227–242.

[12] K. Dmitriev, J. Schumann, and F. Holzapfel, "Toward certification of machine-learning systems for low criticality airborne applications," in *2021 AIAA/IEEE 40th Digital Avionics Systems Conference (DASC)*. IEEE, 2021, pp. 1–10.

[13] ——, "Towards Design Assurance Level C for Machine-Learning Airborne Applications," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*. IEEE, 2022, pp. 1–6.

[14] "EASA concept paper: First usable guidance for level 1 machine learning applications," European Aviation Safety Agency, Tech. Rep., 2021.

[15] "EASA concept paper: First usable guidance for level 1&2 machine learning applications," European Aviation Safety Agency, Tech. Rep., 2023.

[16] "Neural network based runway landing guidance for general aviation autoland," Federal Aviation Administration, Tech. Rep. DOT/FAA/TC-21/48, 2022.

[17] "Concepts of design assurance for neural networks (CoDANN)," European Aviation Safety Agency, Tech. Rep., 2020.

[18] "Report. concepts of design assurance for neural networks (CoDANN) II," European Aviation Safety Agency, Tech. Rep., 2021.

[19] *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes*, EASA Std. CS-25 Amendement 28, 2023.

[20] *System Design and Analysis*, FAA Std. AC 25.1309-1A, 1988.

[21] *System Safety Analysis and Assessment for Part 23 Airplanes*, FAA Std. AC-23.1309-1E, 2011.

[22] *Software Considerations in Airborne Systems and Equipment Certification*, RTCA, Inc. Std. RTCA DO-178C, 2011.

[23] *Design Assurance Guidance for Airborne Electronic Hardware*, RTCA, Inc. Std. RTCA DO-254, 2000.

[24] *Certification Specifications for All-Weather Operations*, EASA Std. CS-AWO Issue 2, 2022.

[25] *Annex 10 - Aeronautical Telecommunications - Volume I - Radio Navigational Aids*, ICAO Std. AN 10-1, 2023.

[26] *Minimum Operational Performance Standards (MOPS) for Global Positioning System/Satellite-Based Augmentation System Airborne Equipment*, RTCA, Inc. Std. RTCA DO-229, 2020.

[27] K. Dmitriev, J. Schumann, and F. Holzapfel, "Toward Design Assurance of Machine-Learning Airborne Systems," in *AIAA SciTech 2022 Forum*, 2022, p. 1134.

[28] K. Dmitriev, J. Schumann, I. Bostanov, M. Abdelhamid, and F. Holzapfel, "Runway sign classifier: A dal c certifiable machine learning system," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*. IEEE, 2023, pp. 1–8.

[29] *Standards for Airport Sign Systems*, FAA Std. AC 150/5340-18G, 2020.

[30] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv e-prints*, pp. arXiv–1804, 2018.

[31] ——, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[32] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.