frontiers | Frontiers in Astronomy and Space Sciences

TYPE Review
PUBLISHED 15 November 2022
DOI 10.3389/fspas.2022.896877

Check for updates

OPEN ACCESS

EDITED BY
Bala Poduval,
University of New Hampshire,
United States

REVIEWED BY
Marie Farrell,
Maynooth University, Ireland
Verena Heidrich-Meisner,
University of Kiel, Germany

*CORRESPONDENCE
Haroun El Mir,
H.El-Mir@cranfield.ac.uk

SPECIALTY SECTION
This article was submitted to Space Physics,
a section of the journal
Frontiers in Astronomy and Space Sciences

RECEIVED 15 March 2022
ACCEPTED 02 November 2022
PUBLISHED 15 November 2022

CITATION
El Mir H and Perinpanayagam S (2022), Certification of machine learning algorithms for safe-life assessment of landing gear.
Front. Astron. Space Sci. 9:896877.
doi: 10.3389/fspas.2022.896877

COPYRIGHT
© 2022 El Mir and Perinpanayagam. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Certification of machine learning algorithms for safe-life assessment of landing gear

Haroun El Mir* and Suresh Perinpanayagam

Integrated Vehicle Health Management Centre, Cranfield University, Cranfield, United Kingdom

This paper provides information on current certification of landing gear available for use in the aerospace industry. Moving forward, machine learning is part of structural health monitoring, which is being used by the aircraft industry. The non-deterministic nature of deep learning algorithms is regarded as a hurdle for certification and verification for use in the highly-regulated aerospace industry. This paper brings forth its regulation requirements and the emergence of standardisation efforts. To be able to validate machine learning for safety critical applications such as landing gear, the safe-life fatigue assessment needs to be certified such that the remaining useful life may be accurately predicted and trusted. A coverage of future certification for the usage of machine learning in safety-critical aerospace systems is provided, taking into consideration both the risk management and explainability for different end user categories involved in the certification process. Additionally, provisional use case scenarios are demonstrated, in which risk assessments and uncertainties are incorporated for the implementation of a proposed certification approach targeting offline machine learning models and their explainable usage for predicting the remaining useful life of landing gear systems based on the safe-life method.

KEYWORDS

explainable AI, landing gear systems, certification, risk management, safe-life design

1 Introduction

The aircraft maintenance, repair and operations (MRO) industry is seeing a rise in demand for new aircraft, as well as an increased need for seamless integration and cost-effective maintenance digitisation. Digital, or avionics systems, are rooted as a progressively-important part of the predictive maintenance processes used in aircraft. Examples of such systems are a division of structural health monitoring (SHM), named damage monitoring systems. It consists of load monitoring, also known as operational loads monitoring (OLM), and fatigue monitoring (Staszewski and Boller, 2004). With the advancement in processing power, computing capabilities of onboard systems are rendered able to effortlessly accommodate improved and more demanding loads monitoring sensors and software. This paper explores the improvement of fatigue monitoring systems for landing gear (LG). LG are certified for usage on aircraft using

the safe-life fatigue approach. This approach attributes each component of the LG with a predefined and unchanging service life, after which the component is either:

1) Used as a replacement to a similar component onto the LG assembly of another aircraft, wherein it is certified for a longer life span due to the less impactful load profile on the aircraft in which it will be used.
2) Scrapped and deemed unworthy of service.

The safe-life calculation consists of a load spectrum assigned to the aircraft LG, which consists of an assumption that forms a safety factor. This load spectrum estimation can accommodate improvements, due to its high safety factors. The loads applied in-service are highly probable to be less impactful on the life of the part than what is proposed by the safe-life estimation. The assigned service life may therefore be extended if the loads are monitored with OLM equipment. The disparity in stress-life (S-N) curves also contributes to the value of the safety factor applied when setting the safe-life of the component (Irving et al., 1999).

The safe-life method assumes a set of load profiles to result with a number of trips that the LG will be safely able to travel. Instead, basing the replacement of the LG on the amount and severity of loads encountered can be accomplished by collecting data with the use of sensors, thereby allowing for the quantification and classification of the factors causing imminent fatigue failure. Currently, such an ideology is approached by a form of OLM systems, which consists of strain gauges placed on military aircraft (Hunt and Hebden, 2001; Dziendzikowski et al., 2021) wherein the strain output is transformed into digital signals that are thereby converted into stress histories, resulting in a loading sequence. Nevertheless, this method of fatigue assessment is inadequate for structural damage detection, by virtue of leaving out "a factor of two to three in fatigue life to be gained if damage could be monitored more adequately" (Staszewski and Boller, 2004). Furthermore, placing additional devices for measuring such parameters invites more reliability issues and an increase in maintenance costs (Cross et al., 2012).

This has, in turn, given birth to the use of Artificial Intelligence (AI)-handled solutions, with an expected growth due to commercial demands, closing the gap where safety-critical applications and the novelty of machine learning (ML) algorithms are deemed to ultimately collide and remould the way that the MRO industry has been assessing aircraft structural health. Successively, the emergency of placing a basis for the certification and risk management of such approaches arises, ranging from the ML explainability levels to the uncertainties in data exchange and collection in-service, due to the non-deterministic qualities of ML when compared to currently-used avionics software and equipment.

Aerospace industry regulators have put forward their interest in the use of ML, for its data-driven benefits, in digital systems related to all levels of the aircraft development cycle, from design to manufacturing, maintenance and operation to communication, by assigning committees and publishing recommendations. EUROCAE created working group WG-114, and SAE started committee G-34, both working in conjunction with the aim of certifying AI for the safe operation of aerospace vehicles and systems, including Unmanned Air Systems. Their published work so far has been the "*SAE AIR6988 & EUROCAE ER-022 Artificial Intelligence in Aeronautical Systems: Statement of Concerns*" (SAE International, 2021a; EUROCAE, 2021). It critically assesses current aeronautical systems encompassing the whole lifecycle of airborne vehicles and equipment and how they fall short of covering AI and, more specifically, ML challenges.

A coverage of upcoming certification requirements for the usage and collection of data from aircraft sensors to predict LG remaining useful life (RUL) is employed in this paper. It is based on:

1) The WG-114/G-34 SAE AIR6988 document (SAE International, 2021a).
2) EASA AI Roadmap (EASA, 2020).
3) EASA CoDANN & CoDANN II reports (EASA and Daedalean AG, 2020, 2021).
4) EASA Concept Paper: First Usable Guidance for L1 ML Applications (EASA, 2021).

These documents have been chosen due to their relevance to the subject of this paper. Nevertheless, the documents also incorporate previous standardisation requirements (such as ARP4754A and DO-178C, DO-254) and guidance by means of addressing their limitations in light of AI requirements for avionics applications. For a survey and taxonomy of the recently-published proposals and guidance papers on practical ML application for use in aviation, the article by the subgroups of the SAE G-34/EUROCAE WG-114 standardisation working group on ML lifecycle development (Kaakai et al., 2022) is recommended to the reader. It sets out the ML development lifecycle guidelines for certification in aeronautics, that are to be the core of the forthcoming publication by SAE: the "*AS6983 Process Standard for Development and Certification/ Approval of Aeronautical Safety-Related Products Implementing AI*".

2 Paper contribution

This paper encapsulates the certification approaches and requirements currently available for landing gear (LG) and AI applications in the aerospace industry, to cover all issues related to machine learning (ML) and safe-life, which will eventually lead

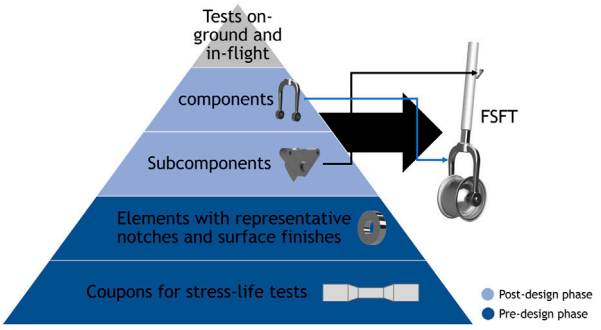


FIGURE 1
The life cycle and process of testing for aircraft components begins, generally, with the pre-design (non-specific) phase, consisting of material coupon and element testing. The process may be compared to stages, or levels, where the testing of each stage, with its corresponding attribute, is required to be completed successfully in order for the next, upper, and more structurally complicated stage to begin. At the bottom of the pyramid, a coupon is stress-tested physically; a specimen of material tested for the nature of its properties. Then, to be more representative of the part used, elements with a similar surface finish and treatment, in addition to including notches characteristic of the part in question, are tested. At the post-design phase, sub-component and component physical testing, which considers the impact of environmental conditions and key features of the LG e.g., piston-cylinder assembly performance, leads to FSFT tests, where the whole structure, or its vital system component sets, such as the shock absorber system, steering system, and wheel and braking system are flight-simulated. Finally, the structure is tested on-ground and in-flight. The process shown in the figure is also known as the "building block" test approach.

to a philosophy for the certification of ML for LG, and whether it may be employed using AI in the next decade. Major issues related to AI that affect the LG environment will have been identified by the reader. It is important to note that the goal of this paper is to illuminate and ease the process of the development of a certification methodology, where a ML algorithm/set of algorithms are to be used for the purpose of LG remaining useful life (RUL) prediction. The paper does so by assisting with confusions a newcomer to this field may have, as the area of certification is quite tough to manoeuvre. The reader may then form a method with which to begin and is guided along the way with the allocation of their requirements through the elimination of current standards and allocation of assurance case tools available, as well as building, block by block, a clearer image of where they stand in the process of complying with those standards, in order to develop adequate use case scenarios.

3 Current Safe-Life Assessment

Safe-life fatigue analysis of aircraft structures is a principle of design in which an estimation is placed prior to the first operation of a component in-service. This estimation is based on the evaluation of the structure's ability to sustain its original crack-free status while being exposed to cyclic loads in-service, such as landing, take-off, and taxiing, which all contribute to impacting the fatigue life of the LG components (Ladda and Struck, 1991). The safe-life analysis places a value of operational hours for the part in question in which it would be replaced afterwards, regardless of whether visible fatigue cracks form in the structure. This approach therefore deems the part inoperable and unsafe for use on the aircraft after those specified hours or cycles. Looking towards how this approach begins, the component's lifecycle and its workarounds, as well as how they fit into the whole aircraft's production plan, come into question.

3.1 Aircraft testing lifecycle

The life cycle and process of testing for aircraft components begins, generally, with the pre-design (non-specific) phase, consisting of material coupon and element testing. The process may be compared to stages, or levels, where the testing of each stage, with its corresponding attribute, is required to be completed successfully in order for the next, upper, and more structurally complicated stage to begin. At the bottom of the pyramid in Figure 1, a coupon is stress-tested physically; a specimen of material tested for the nature of its properties. Then, to be more representative of the part used, elements with a similar surface finish and treatment, in addition to including notches characteristic of the part in question, are tested. At the post-design phase, sub-component and component physical testing, which considers the impact of environmental conditions and key features of the LG e.g. piston-cylinder assembly performance, leads to FSFT tests, where the whole structure, or its vital system component sets, such as the shock absorber system, steering system, and wheel and braking system are flight-simulated. Finally, the structure is tested on-ground and in-flight (Ball et al., 2006). The process shown in the figure is also known as the "building block" test approach (Wanhill, 2018). The post-design phase is directly connected to airworthiness certification due to the phase containing components and parts of the aircraft ready for use and in their final stage of design (Ball et al., 2006). Compliance with airworthiness standards demands the identification of loads encountered and the load cycles in order to schedule corresponding component visual check-ups (Wong et al., 2018).

3.2 Safe-life requirements

Currently, the only components in the aircraft to which this safe-life fatigue estimation may be applied are the LG. The LG's incapability of accommodating crack initiation and expansion is due to its components consisting of high-strength alloys that

motivate rapid crack propagation. Two fatigue detection approaches may be used for the safe-life fatigue analysis of a metallic aircraft component: the stress-life approach and the strain-life approach (Wanhill, 2018). The ways in which a safe-life is specified to obtain certification allowing the use of the component on large aircraft requires:

1) Full-scale fatigue tests (FSFT) encompassing the whole structure physically being tested with methods, such as strain gauges mounted to localise and quantify strain (Dziendzikowski et al., 2021).
2) The testing of specific components of that structure in question–in the case of LG, that would be its individual components each tested separately for fatigue resistance.
3) The use of hypotheses and the stress-life approach *via* Miner's rule for damage accumulation, whereby damage fixated by each repetition of stress due to load applications is assumed equal (Federal Aviation Administration, 2005). The Miner's rule, also referred to as the Palmgren-Miner linear accumulation hypothesis, states that the damage due to fatigue is equal to a singular value of "one" as long as cyclic application of this load has reached an amount validating its appearance on the fatigue curve (Schmidt, 2021).

The LG encounters multiple loads in succession, contributing to high cycle fatigue (HCF). Low cycle fatigue, which is correlated with strain life curves, is characterized by plastic strain. Stress-life curves, on the other hand, are used in high cycle fatigue, where fatigue is mostly in the elastic region and plasticity can be neglected. Landing gear stresses do not reach the plastic deformation region of the material in each of its components, which is why the stress-life fatigue approach is used. There is an abundance of available data for the stress-life approach, and it is applicable specifically to HCF. In addition to the pre-design nature of the landing gear structural CS-25 airworthiness certification requirements for large airplanes, the safe-life fatigue analysis that is currently used in the LG certification process utilises Miner's rule for damage accumulation, using S-N curves. These curves conform to a certain material coupon, where the material must be the same as that used in the component in question. As Pascual and Meeker (1999) discuss, an S-N curve for a certain material is a representation of the fatigue data of a coupon of that material, in the form of a log-log plot containing cyclic stress 'S' values *versus* 'N', the median fatigue life articulated in cycles to failure. It is key to note that S-N curves are derived from a specific stress-ratio. They also contain scatter, which is an uncertainty associated with failure in fatigue. Additionally, two factors parametrise S-N curves: probability of survival and probability of failure. Both introduce uncertainty factors to be applied for the final prediction of a component life. Fatigue is non-deterministic, as opposed to static loads, e.g. Component A tested for fatigue using the identical test parameters as Component B will result with a fatigue life significantly different than that of its proponent. This introduces scatter in S-N curves used for fatigue prediction.

Required in addition to these curves is the fatigue spectrum: data on the applied loads, how frequently they manifest, and how their occurrence fits in the grand scheme of load sets applied, in terms of their timing and repetitions. Flight profiles are a set of load variances, representative of a certain flight block. These profiles add up to form a spectrum for fatigue prediction (Schmidt, 2021). The spectrum may also consist of flight hours in addition to flight cycles if the nature of the mission of the aircraft is mixed in terms of range duration. Established design lives can be divided into three categories with their corresponding cycle ranges:

1) 50,000 cycles for short-haul flight aircraft, e.g. A320.
2) 25,000 cycles for long-haul aircraft, e.g. A350.
3) 10,000 cycles for tactical aircraft.

The steps for safe-life fatigue analysis of LG are as follows, summarised in Figure 2:

Step 1. S-N curves are generated by performing uniaxial cyclic stress amplitude loads on numerous material samples until failure. This material data may also be extracted from readily-available scatter data and must comply with the 99/95 standard, with an applied scatter factor of 3 at a minimum (Fatemi and Vangt, 1998) The curves are also altered according to the in-service factors of the landing gear environment, which are not experienced by the coupons tested in monitored conditions. The resulting curves are referred to as "working curves" (Wanhill, 2018).

Step 2. Meanwhile, a stress-time history plot is derived from a load-time history plot for the LG component by referring to the geometry of the component.

Step 3. Methods such as Bathtub/Rainflow counting are performed on the load-time history plot which is a stress-time history plot after Step 2, to result in stress cycles and a mean stress value for each cycle count (Le-The, 2016).

Step 4. The cycles are converted with their mean values to fully-reversed stress cycles in order to extract equivalent data when referring to the S-N curves for the material used in the component of the LG (for data compatibility purposes). This is done *via* mean stress correction techniques, such as the Goodman mean stress correction Eq. 1. (σ_0), as discussed by Hoole (2020), is the value of the fully-reversed stress cycles. (σ_a) is the stress amplitude value of those stress cycles, (σ_m) is their mean stress level, and (σ_{UTS}) is the material's defined ultimate tensile strength.

Step 5. Fatigue damage (d) accrued by each applied cyclic stress amplitude (σ_0) is formulated using Miner's rule. As per Equation 2 (n) is the frequency at which (σ_0) is applied, and (N_f) is the number of cycles to failure. (D_T) is total damage

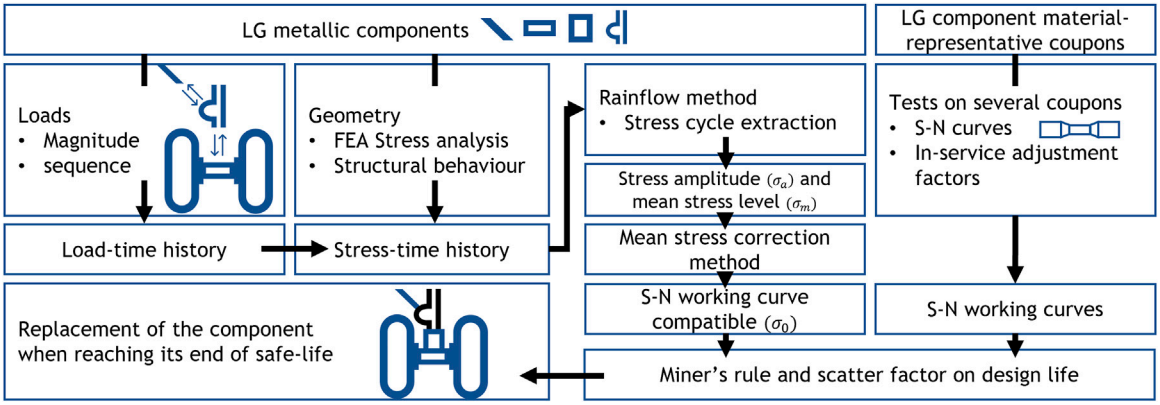


FIGURE 2
Steps for safe-life fatigue analysis of landing gear.

accumulated from the stress formed by the cycles. As (Hoole, 2020) mentions, a value of 1 for (D_T) signifies failure of the part in question, meaning it has reached the end of its fatigue life, and representing failure Eq. 3.

Equation 1 Goodman mean stress correction

$$\sigma_0 = \frac{\sigma_a}{1 - \frac{\sigma_m}{\sigma_{UTS}}} \tag{1}$$

Equation 2 Fatigue damage, Miner's rule

$$d = \frac{n}{N_f} \tag{2}$$

Equation 3 Total damage, Miner's rule

$$D_T = \sum d \tag{3}$$

3.3 Machine learning for safe-life prediction

Studies performed by Holmes et al. (2016) attempt to form a correlation between flight parameters and loads applied to a LG structure attached to a drop test rig, *via* the use of two types of nonlinear regression models as part of their ML approach: multi-layer perceptron (MLP), and Bayesian MLP. The data accumulated consists of inputs, such as wheel speed, accelerations in the LG, and similar flight variables, consisting of kinematic approaches; related with changes in velocity and displacement, in order to result in load induced on the LG. Since the MLP is Bayesian, it requires a specification of a prior. A gaussian prior distributions was used. The functional efficiency of the used neural network (NN) is calculated by acquiring the mean-square error between the predictions formed by the model and the measured targets. Optimising the NN is done using gradient descent. As for the weight uncertainty of the NN, it is reduced by assigning each weight a probability distribution. Additionally, the input datasets were filtered due to noise in acceleration measurements being higher than actual load values recorded through strain. The physical test of the LG rig included assumptions made to simulate a landing environment *via* spinning the wheels before impact, changing the angle of impact of the LG, and dropping the structure from variable heights. These impacts were then measured using strain gauges placed on the LG rig components and load cells placed on the platform on which the LG drops. Another method used for data collection and prediction included the use of Greedy algorithms and Gaussian process (GP) regression; a class of Bayesian non-parametric models. With the use of flight test data parameters to predict landing gear vertical load. GP was used as it trains faster than MLP, and the computations necessary for GP regression are simplified by the fact that a distribution directly over candidate functions can be defined, rather than over the parameters of a predefined function (as would be necessary for a Bayesian neural network for example). They are likewise compact. Cross et al. (2013) found correlations with the general trend of data prediction. Later studies put forth the requirement of physics-informed data to predict landing gear loads to a usable level. These ML approaches result in models that are able to predict loads, where a model requires that it be aircraft-specific. Nonetheless, different surfaces on which the physics-informed ML model (using both LG drop test data and flight test data) was used on still produced acceptable outcomes.

4 Machine learning techniques

As a branch of AI, ML is a computing field that operates with the use of computational methods related to statistics, probability, and computing theory. ML is used by systems to

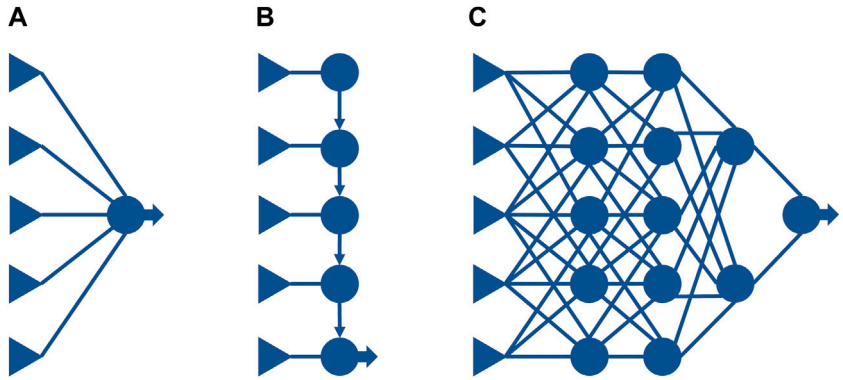


FIGURE 3
Neural networks in comparison to linear regression and decision lists. A linear model **(A)** such as linear or logistic regression is able to compute and take in a high number of variables for input. Nevertheless, the path from input to output is relatively short due to all variables being multiplied by a single weight, in addition to the principle that these input variables are not capable of communicating within themselves. This renders them able to only act for linear functions and boundaries related to the input space. Decision lists **(B)** allow for these long paths of computation to occur, but depends on the input variables being of a similar size to the output variables. Neural networks **(C)** merge these two methods together, allowing for the input variable interactions to be complex and incorporate long computation paths. The benefit of this model is the ability to represent applications, such as speech, photo and text recognition.

learn patterns or monitor data input and apply statistical algorithms to infer the required output depending on the type of algorithm being used. The method by which models of ML operate may be described as follows: "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, measured by P, improves with experience E" (Mitchell, 1997). An example is the use of statistical methods, where algorithms classify or foresee similarities in data being extracted to suggest a best-case scenario. The input data used to build the ML model, through the stages of its creation, are categorised into three common datasets, forming the ML algorithm: training, validation and testing. Furthermore, ML may be categorised into four types in terms of the method with which it learns: supervised, unsupervised, semi-supervised and reinforcement learning. These reflect the types of feedback-input relationships. Supervised learning occurs when input and output pairs of labelled data are monitored and a function is learned as a result, mapping input and output accordingly. In unsupervised learning, the unlabelled data input is studied without any feedback and patterns are found within that input. Semi-supervised learning trains on labelled and unlabelled data, improving model accuracy when compared to a supervised learning algorithm. As for reinforcement learning, the algorithm is given a response at the end of each set of decisions made, as part of each step in its decision process. Its aim is twofold: the initial improvement of performance due to learning from previous action-result combinations, and the eventual output of the most optimal long-term reward that it may be assigned, e.g. lengthening the duration of a game in order to win eventually instead of winning over an opponent earlier on only to ultimately lose in a game of checkers (Russell and Norvig, 2022).

4.1 Artificial neural networks

When ML involves layers of computing segments that are adaptable and unembellished, that is the term known as deep learning. Deep neural networks (DNN), a subset of ML, are the most common form of deep learning. They are based on one or more layers adapted for large data input sizes. When containing less than 3 layers, the term neural networks is used. Figure 3 is a demonstration of how a DNN may relate to shallower ML models. A linear model (a) such as linear or logistic regression is able to compute and take in a high number of variables for input. Nevertheless, the path from input to output is relatively short due to all of the variables being multiplied by a single weight, in addition to the principle that these input variables are not capable of communicating within themselves. This renders them able to only act for linear functions and boundaries related to the input space. Decision lists (b) allow for these long paths of computation to occur, but depends on the input variables being of a similar size to the output variables. Neural networks (c) merge these two methods together, allowing for the input variable interactions to be complex and incorporate long computation paths. The benefit of this model is the ability to represent applications, such as speech, photo and text recognition (Russell and Norvig, 2022). DNN, which are characterized by multiple layers instead of one (usually three

layers or more), tend to be more accurate and effective in task purveyance. A term commonly found when dealing with the inexplicability of DNN, the black-box is scientifically associated with a system of known and observable inputs and outputs, and no knowledge or observation to be made on how the inner mechanisms of that system may be. In the case of a NN, although the code may be observed, it is functionally referred to as a black-box due to the nature of constant reorganization of the computational NN layers. Modelled based on the workings of the brain; firing neurons with correlated weights to result with decisions, the black-box model and nature of DNN has recently been subjected to theories attempting to explain its method of operation, some attempting to generalize to all types of DNN modes of operations (Alain and Bengio, 2016; Zhang et al., 2016; Schwartz-Ziv and Tishby, 2017; Poggio et al., 2020), and others focusing on certain NN methods and the available interpretation approaches (Guidotti et al., 2018; Montavon et al., 2018; Azodi et al., 2020).

4.2 Machine learning challenges

In health monitoring of aerospace structures, an advisory system provides recommendations that are backed up with evidence, which are in the form of:

1) Sensor output from damage monitoring systems, which consists of direct measurements from the aircraft component/s in question.
2) Flight parameter and environmental conditions derived outputs, that are indirect measurements. These materialize in the form of operational monitoring systems (OMS).

The OMS is a sub-component of SHM, and a system similar to damage monitoring, with the difference being that its measurements are of a derived nature (SAE International, 2021b). The former is the system most useful for the purpose of sensor replacement purposes. Nevertheless, ML may be used as a part of both damage monitoring and OMS. Requirement-wise, the software that provides an envelope around the ML tool needs to be developed to a defined quality process, according to a distinct software control method. That occurs when embedding the software. In addition, it must be demonstrated that the ML black-box may be used in a reliable and robust manner. Questions important for the setting of requirements in the ML uncertainties capture are:

1) 'Is it using recognized libraries?'; code pre-written for repeated usage. The reader is referred to (Nguyen et al., 2019) for a description and comparison of current ML libraries and frameworks.
2) 'What was the quality process used in creating that software?' A framework by Murphy, Kaiser and Arias (2006) proposes a ranking for supervised ML algorithms, consisting of "tools to compare the output models and rankings, several trace options inserted into the ML implementations, and utilities to help analyse the traces to aid in debugging".
3) 'What is the validation process of the model itself (the data-driven part of the training)?'

Just as important, the training, testing and validation processes must be robust and contain a level of assurance that provides accurate predictions when implemented live, in order to be moved from an advisory status to a fully-trusted status. What data is used, its source, reliability, coverage provided by the data (e.g., whether it covers all types of landing for the aircraft type in question), and all operational cases (e.g., heavy landing, light landing, crosswind conditions, icy conditions on runway) are questions to be asked when formulating a data-based rigorous selection process. Moreover, whether the validation data is based on physics data from finite element (FE) models, or testing rig scenarios, plays a significant role in the assurance process.

Deep learning encounters challenges pertaining to its data in which features are represented, specifically with the initial step of obtaining that data, wherein labelling is required (Khan and Yairi, 2018). Furthermore, challenges introduce themselves, according to Khan and Yairi (2018) in the following aspects and identifiers of the deep learning bubble:

1) Specific deep learning architectures and their categorisations into the most suitable pertaining applications have not been yet solidified due to researchers' inadequate justifications of why they used those specific methods and as to why a certain number of layers was most suitable for their applications.
2) Comparison of the architectures has not been standardised, whether it be in terms of time consumption, resource management, computational requirements, or data loss.
3) With regard to structural health management, deep learning applications will have to recognise the failures or faults according to their corresponding environments and be able to diagnose issues, such as no fault found (Khan et al., 2014a; 2014b).

Of the problems faced, imbalanced data issues arise. As discussed by Liu et al. (2009), the cause of imbalanced data results while learning is due to classification and clustering situations, as a result of the classes being learned having considerably more data when compared to their counterparts. Furthermore, cases which are uneven occur due to the intrinsic nature of those events, as well as the additional expense that may result from obtaining these examples for learning in the algorithm. These imbalanced data classification issues may be overcome with the following approaches: pre-processing, cost-sensitive learning, algorithm-centred, and hybrid methods (Kaur et al., 2019).

The data used for training an algorithm may be improved with pre-processing methods, when the algorithm faces a class of data containing an abundant number of examples while the other class contains a lower amount. Due to the accuracy of classification being negatively affected if not for sampling methods, they represent an important step towards avoiding bias (Barandela et al., 2004). The aim of these methods is to balance the classes of data and result with less bias *via* either over-sampling or under-sampling. These two methods operate by manipulating the training data space.

Over-sampling: Of the classes available in pre-processing data, the minority class that happens to bias the data is duplicated in sample packets and the data is therefore balanced in terms of the final dataset. Under-sampling, on the other hand, performs the opposite by randomly extracting samples from the major class (leading to the probable negative aspect of deleting important data) in order to result in equal amounts of the minor and major class. Over- and under-sampling may be combined to form the hybrid sampling method, where they are both used to result with balanced data for pre-processing (Xu et al., 2020).

Bias and variance are concluded to be the key issues in ML applications. They are to be addressed, according to (EASA and Daedalean AG, 2020), based on the following two methodologies:

1) Datasets with bias and variance need to be distinguished from opposing datasets and effort shall be put into reducing such bias and variance within the data itself.
2) The bias and variance need to be evaluated based on the level of risk they impose upon the ML model.

Feature selection and extraction are another means of selecting features more suitable for the classification at hand at the pre-processing stage (Kursa and Rudnicki, 2011). The classes of feature selection would be the filter method, wrapper method, and embedded methods (Guyon and Elisseeff, 2003).

4.3 ML risk management

A ML workspace is a framework in which the algorithm's training takes place. The workspace allows for the specification of the coding language package to be used, its training preferences, and the workspace variables. According to SAE AIR6988, as part of the advised requirements to forming certification standards for the data selection and validation of ML systems, the workspace should be covered with a certain level of protection to prevent "data poisoning or tampering", whether it be intentional or not, by the workspace user or intruder. The effects of such an intrusion would include false outputs and algorithm decisions, e.g., importing additional data into the training dataset which cause the algorithm to develop a deceptive result while assuming that the training process is untampered with. Moreover, any non-complying data must be detected and removed from the dataset after the validation step. Additionally, the "probabilistic nature of ML applications" must be taken critically when assessing and forming the safety process analysis.

For the certification of the method in which data is selected and validated, validation for ML would partition a block of data, representing the entire operational profile of a landing system, into 3 types:

1) Training, in which the model in this cycle is trained and compared with the results from an independent dataset which would be the validation set, and a decision is formulated: is this model good enough or does it require further refinement? This decision set is part of the training cycle, clarifying the need for the validation set to be independent of the training set.
2) Testing, where each of these datasets needs to conform with IID (Independent Identity Distributed) and be of good coverage. For example, in a scenario where hard landings are part of the data input, a similar number of hard landings in each of those three datasets must be clearly present in order to avoid the inevitability of bias.
3) The validation process of the model itself, in which the safe-life approach for LG RUL assessment would be the benchmark for this paper's purposes. The model's performance in this step is evaluated by means of using the validation dataset (set aside and unused, as part of the data partitioning procedure done beforehand) and observing the output to decide whether it is acceptable, signalling the readiness of the ML algorithm for use in a real-life scenario, if so.

Risks in ML are categorised, in terms of robustness, into two kinds (EASA and Daedalean AG, 2021):

1) Algorithm robustness, where the algorithm used for learning is tested for robustness as the training dataset is changed.
2) Model robustness, in which perturbations in the input to the algorithm are used for the identification and quantification of the robustness of the training model.

As pointed out in AIRC6988, the traditional form of safety assessment has always been to realise the orders of system failure by means of its own component-level intercommunication with other systems. This could be improved for the case of AI applications due to their complicated ecosystem interactions. The interaction of the system with "external factors" is one improvement to be noticeably important, due to its probability of forming failure conditions in the case of AI applications. Such a safety approach already exists as part of the SOTIF_ISO 21448 document for certification based on the automotive industry's "advanced algorithms" system inclusions. This approach assesses the following:

TABLE 1 RUL ML black-box issues and proposed mitigations.

RUL ML black-box issue	Corresponding mitigation
ML models need to cater to the varying nature of fatigue life scatter in data points in order to appropriately "characterise the probabilistic property of fatigue lives given a specific condition"	Certification requirements must capture fatigue life scatter data and be able to predict fatigue life probabilistically
ML models learn correlations between data input and output via the means of data extraction, leading to the possibility of contradicting physics principles	Certification requirements must adapt to models trained with different datasets
Using a model trained on one data range may result inaccurately when the same model is implemented on a different framework due to the possibility of data overfitting	Certification requirements need to incorporate vital landing gear operational uncertainties, such as hard landings, as well as temperature and environmental variations

1) Both the system and sub-system levels of AI are tested for functionality and performance.
2) The probable sources of failures mitigated by the functional aspect of the system must be pointed out and their causes reassessed.
3) These probable failures must be avoided by the means of "functional modifications".

Putting these advisories into effect, in the case of issues that will arise due to the usage of black-box ML models in order to model fatigue life, an advisory example is shown in Table 1. A high-level mitigation, or requirement, is set up for each ML data issue.

Certain ML infrastructures, such as continual learning pipelines, allow for the ability to add continuous data points in a well-formulated algorithm, allowing for the data output to be optimised in terms of the assessment of structural integrity and maintenance scheduling. This is deemed an improvement for data collection purposes, but increases the risks for uncertainties specifically when considering external data collection factors, where the potential sources of data in the case of LG fatigue detection include:

1) Fatigue tests implemented physically on the parts themselves in a controlled environment.
2) Flight data of the same aircraft and landing gear from other operators.
3) Maintenance observations.
4) IVHM data, including output from strain gauges on-board the aircraft and LG assembly.

4.4 Explainability

Certification for ML applications in LG may be applied *via* explainability, by the means of connecting data point values from features; values and properties of a monitored process (Bishop, 2006). Among the important requirements for the acceptance of a ML algorithm for use in an industry that is to accept AI solutions over the coming years, trust reappears as a main question at hand, which is where explainability comes into play. Applications and methods for instilling trust into a certain AI approach are reflected in the currently-adopted Intelligence Community Directive (ICD 203) and the SAE AIR6988 documents. These both serve the purpose of proposing the standards required for the application of AI in the aerospace industry, as well as emphasising the need for explainability (Blasch et al., 2019). Additionally, explainability is a part of the four building blocks of the framework in EASA's guidance for ML applications paper (EASA, 2021), in addition to the DEEL white paper (DEEL Certification Workgroup, 2021) that concentrates on the properties an ML system should have, and specifies those to be "auditability, data quality, explainability, maintainability, resilience, robustness, specifiability, and verifiability" (Kaakai et al., 2022).

Explainability is a method by which the transparency of a ML black-box may be improved, where the ML model being explained gets its model prediction uncertainties specified by the user, as well as the clarification of the method with which the feedback of the model is interpreted takes place. Such explainable methods have already been achieved by the means of the research done by Smith-Renner et al. (2020):

1) Ensuring fairness in the model with which the end users may interpret the meaning of the results in a language that conforms with their own specific knowledge and terminologies, while assessing bias in the meantime (Dodge et al., 2019).
2) Adjusting the expectancies of end users to comply with the end results of the explainable AI method being used in which uncertainties in the ML model itself are incorporated for the user to be prepared in terms of the model perception (Kocielnik et al., 2019).
3) Enclose trust of an explainable AI agent in order for users to return to such an ML algorithm repeatedly for similar use case scenarios encompassing the model's features of its system, its agents' reliability, and the intentions with which trust is to be instilled (Pu and Chen, 2006).
4) Improve the recommendation rigor of the explainability of the black-box ML model by means of clarifying to the user

which parts of the model are the most important for the use case scenario at hand while referring to the conceptual model in the user's mindset (Herlocker et al., 2000).

Furthermore, explainability may be organised in regard to its approach to the ML algorithm, in which feature selection and feature extraction are two distinguished methods. Feature extraction creates non-detectable features from those that have already been found in the algorithm (Guyon et al., 2006),whereas feature selection evaluates each and every feature in the model after which these features are deemed either adequate or inconsistent for use in the model (Guyon and Elisseeff, 2003).

Another term important to the explainability approach is whether it is local or global in reach. If local, when provided with a conditional distribution, the input clusters of small regions of that distribution lead to how the ML model's predictions are interpreted by the explainable method. As for the case of it being global, average values are the lead source taken for interpretation of distributions fully encompassing the model's condition (Hall et al., 2017).

The method of adopting a ML model's features depends on both the ML model being used, as well as the fatigue failure model being implemented, resulting in the dependence on the sensor data taken ultimately during flight, take-off, and manoeuvres on the landing strip. The usage of features has been resorted to due to the nature of the way in which an ML model operates; by operating on 'single values per case' (Ten Zeldam, 2018). As the ML model formulated to operate on failure diagnosis trains on maintenance data and usage data, while simultaneously filtering outliers, and labelling each feature for the readiness of the model, these labelled features will then need to be categorised based on their relative importance to the fatigue failure of the LG components being studied. These values are compared to predefined value ranges that dictate whether a component's stress reactions qualify it as leading to fatigue failure due to the likely repeatability of this value and its cycling resulting in a HCF failure. The values shall include tyre wear, side-stay loads, impact loads, shock absorber travel distance, as well as distance travelled by the wheel, in addition to forces applied on the axle of the LG. The features are then transferred to classes, or diagnoses (Ten Zeldam, 2018). This methodology does result in relative feature importance, informing the end user of how critical a feature is by relating its likelihood of occurrence to the results of a simulated model.

The need for explainability in certification-required applications is bringing forth work such as that by Viaña et al. (2022), where an algorithm is formed of explainable layers; using clustering for parameter initialisation, overcoming state-of the-art algorithms when it comes to fuzzy system-based combinations.

5 Certification and its challenges

Commercial avionics systems and equipment are composed of software and hardware components, developed to comply with their corresponding design standards. These standards are covered by the two leading documents that the FAA and EASA certification authorities refer to for the approval of the systems in question: DO-178C/ED-12C for the compliance of avionics software development with airworthiness requirements (RTCA, 2012), and its complementary document, DO-254/ED-80 for the design assurance of avionics equipment, consisting of both hardware and software (RTCA, 2000). These documents introduce an iteration of design assurance levels (DAL) that are also used in other avionics certification requirement documents, such as ARP4754A. DAL are measures assigned to each function in the avionics system of an aircraft, be it software-based in the case of DO-178C or hardware based for DO-254. The values of these functional measures range from A to E in alphabetical order. They correspond to cases of catastrophic effect to those of no safety effect on the operation of the aircraft, any form of overload on the crew, and therefore the safety of both (Fulton and Vandermolen, 2017). ARP4754A separates DAL into two: FDAL, function development phase, for aircraft functions and systems, and IDAL, item development phase, for electronic hardware and software items. The FDAL process assigns assurance cases ranging from A to E severity levels for functions which are allocated to items in a system. IDAL then assigns assurance levels for each item that is a part of the function in question, as part of electronic hardware or software. Detailed analyses examples may be read in ARP4754A (SAE International, 2010).

Also emerging are assurance cases toolsets, such as AdvoCATE, developed by Denney et al. (2012), offering an alternative to the manual labour of creating safety cases, and their linkage graphically with similar case scenarios, thus reducing time by providing available risk and hazard options, along with the assigning of requirements whether they be high or low level, in a seamless manner. Furthermore, fragments of the sources of documents for the assigned assurance cases can be linked to each correlated node, creating an easily exportable diagram, the software also works in coordination with AUTOCERT, a tool that evaluates modelling-and-design-stage flight and simulation code for safety violations, *via* clarifying it in a form of wording for the purpose of certification (Denney and Trac, 2008). "*Guidance on the Assurance of Machine Learning for Use in Autonomous Systems*" (AMLAS) is provided with a tool offered by the Institute for Safe Autonomy at the University of York. It focuses on the development of assurance cases for the use of ML in autonomous systems. The tool enables the addition of objects for each ML component, and its corresponding safety cases, while referring to AMLAS detailed means of compliance (Hawkins et al., 2021).

The limitations of ML algorithms require a scope to be identified within, and since they can handle non-deterministic

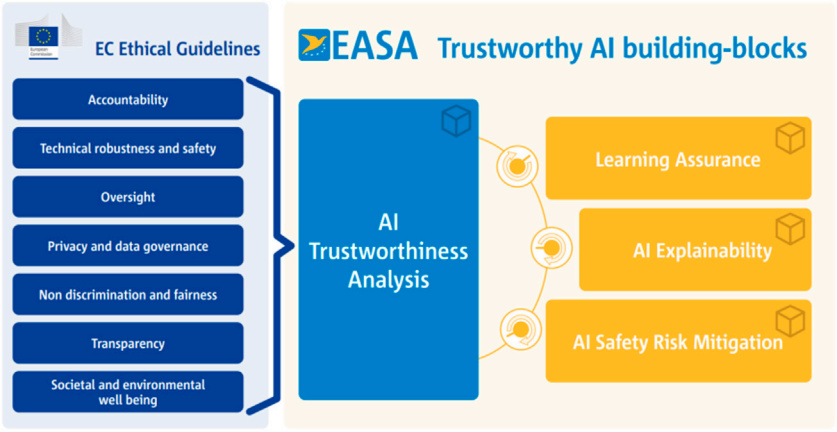


FIGURE 4
EASA Ethical Guidelines and AI building-blocks for trustworthiness. As per EASA's AI Roadmap, which has been formed with the goal of placing standards for ML applications in the EASA-related aerospace sector, seven ethical guidelines were placed for the operation of AI deemed trustworthy. They are subsequently managed by the four blocks in the figure, wherein: AI Trustworthiness Analysis supports the methodology on how to approach the seven guidelines in the use case of civil aviation. Learning Assurance develops the ideology of making sure that the ML algorithm in use is appropriate for the case at hand. AI Explainability focuses on the reason behind why the algorithm decides and its importance with respect to the end user in terms of delivering the desired output. AI Safety Risk Mitigation highlights the nature of how an AI black-box may require supervision due to its understandability and openness being limited in terms of decisions made.

behavioural scenarios, SOTIF (International Organization for Standardization, 2022), which was developed to address the new safety challenges that autonomous (and semi-autonomous) vehicle software developers are facing, may be used as part of the basis for certification application. Another challenge for certification is the constitution of a dataset, and whether it be sufficient for the required application and when compared to the function in operation. In the case of explainability, the lack of such a measure affects confidence in the model's learning capability. While ML is being implemented, the deployment of such a program would not be successful when supplied with a low-level set of tools for the inference. New practices in the aeronautics domain for certification encompass an initiative known as overarching properties. Here, assurance cases, which have been previously used in aeronautics and NN, may define themselves as the bridge between the need to comply with the overarching properties (which are intent, correctness and innocuity) and the quality possession of the product being considered by placing a strong argument. Artificial Intelligence in Aviation workgroups (such as SAE G-34/EUROCAE WG-114) are experimenting with the aforementioned new practices in order to produce guidance material for the standards being developed for ML in the aeronautical domain.

As per EASA's AI Roadmap, which has been formed with the goal of placing standards for ML applications in the EASA-related aerospace sector, seven ethical guidelines were placed for the operation of AI deemed trustworthy, as can be seen in Figure 4. They are subsequently managed by the four blocks in the figure, wherein:

1) AI Trustworthiness Analysis supports the methodology on how to approach the seven guidelines in the use case of civil aviation.
2) Learning Assurance develops the ideology of making sure that the ML algorithm in use is appropriate for the case at hand.
3) AI Explainabilityes focus on the reason behind why the algorithm decides and its importance with respect to the end user in terms of delivering the desired output.
4) AI Safety Risk Mitigation highlights the nature of how an AI black-box may require supervision due to its understandability and openness being limited in terms of decisions made.

5.1 Load profile uncertainties and risk management

Risk management for aircraft commences with following the standards placed by regulatory bodies, such as EASA for the European market, and the FAA in the US market. The next step in uncertainty management would be the categorisation of failure events and their probabilities, wherein there exists an inverse relation between the failure condition of an aircraft and its probability, and the resulting consequence on the aircraft and/or its occupants. Classifications by EASA are defined as Minor, Major, Hazardous, and Catastrophic, where they differ in their

definitions on levels of workload and crew impairment as well as passenger fatality probabilities. In addition, failure types must be stated. These include (Au et al., 2022):

1) Particular Risk: Failures impacting the system from the outside that could affect the system unfavourably.
2) Common Mode: Failure of a component as part of the system that contains a component identical to it dictates that the other component shall fail similarly.
3) Other Isolated Failures: The use of "undetected failures" on systems ensures that a failure not specified explicitly is encompassed in the placed standards and classifications, confirming the robustness of a system, must it pass said introduced diagnosis evaluation without any failure.

The LG operating environment consists of abrupt changes and the electrical sensors are susceptible to such changes and exterior elements. DO-160G covers avionics requirements in terms of environmental test conditions and procedures (Sweeney, 2015). For LG, these include waterproofness, shocks and vibrations, brake temperature, atmospheric conditions, lightning, electromagnetic emissions and susceptibility, and contaminants, such as dust and sand (Au et al., 2022).

A LG's components must be all tested against a "qualification test plan" to prove its usability in the harshest of environmental conditions (Au et al., 2022). This does not, however, include the component's entire life's combinations, resulting with the need to add experience from the industry and a "system development process" to add to the system's decisions in terms of verification for its use-case on-site.

Uncertainties resulting from the fatigue design process may be realised in:

1) Material properties of the components.
2) Geometry of the components.
3) Loads applied in-service onto the components.

The process in which components are manufactured, e.g. machining results with variations in the dimensions of the components, thereby directly affecting stress values of the components while in loading (Hoole, 2020). These variations may add up and amount to a failure as was the case with an aircraft nose landing gear strut examined by Barter et al. (1993), failing due to the formation of a fatigue-induced crack, as a result of an initial defect during manufacturing that grew in-service until the part was overloaded. As for material S-N curve datasets used for the stress-life approach, they naturally contain variability for each stress amplitude when compared to the cycles to failure. Furthermore, during the aircraft manoeuvres, the changes in magnitudes of the loads being applied, as well as when these loads occur, and the order of these occurrences, are factors to be considered for uncertainties. These are overcome *via* the use of safety factors within the stress-life analysis.

Loads imposed on the landing gear as part of the aircraft's life cycle can be divided into two types:

1) High and unexpected landing loads that occur during the aircraft's manoeuvres on the ground e.g. touchdown (Tao et al., 2009).
2) Loads that are repeated during the designated aircraft's trip and while on the ground, e.g. turning, braking, and taxiing, and being towed.

When extracting data, the order in which landing gear loads are applied may be inferred from load-time histories using open-source data, e.g. Flightradar24 such as in the case of Hoole (2020). He further categorises this variability in-service into the following: magnitude of the load, number of manoeuvres on-ground, and the order in which these manoeuvres occur. The latter two depend on factors related to the airport's structure and design, as well as the weather conditions on the day of service, in addition to the aircraft traffic at that point, changing the manoeuvres for an aircraft, also based on each airport's taxi operations locally, as well as gate locations.

For fatigue analysis, and with referral to EASA CS-25, (Hoole, 2020) mentions six methods that are commonly used for RUL conservatism:

1) Safety factors placed on components directly impacting their safe-life in order to indicate that they should be used ahead of assumed failure.
2) A safety factor to adjust the Miner's rule as part of the stress-life life approach discussed previously.
3) A safety factor placed on the application of stress on the components in order to assume that they are larger than their actual values.
4) An S-N curve reduction derived statistically.
5) A downwards shift on the S-N curve, causing the assumed stress required in order to reach failure for a certain number of cycles to be decreased.
6) A shift acting to the left on the S-N curve, indicating the assumption of a lower number of cycles needed in order for a part to fail under the specified load.

6 Proposed scenarios

As is the case with applications that would be deemed safety-critical, the following have requirements imposed upon them by learning assurance standards:

1) Datasets that are important for the development of the system.
2) The method and order in which this development takes place.
3) The behaviour of the system while both the development and operational stages take place.

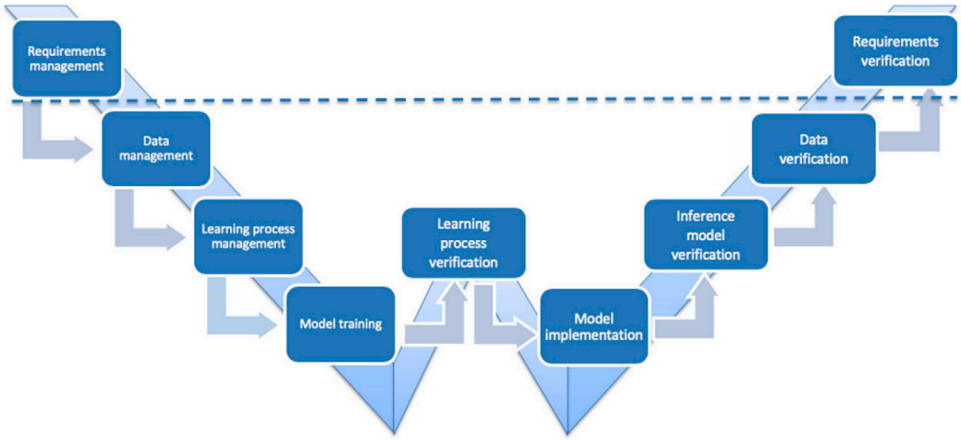


FIGURE 5
EASA Development life-cycle in the case of ML implementation. The process begins and ends with requirements management and verification while taking into reference ED-79A and ARP4754A documentation. In the midst of this life-cycle is data management where training, validation, and test datasets are collected and labelled as well as validated in comparison with the system requirements while sustaining a reliable amount of bias and variance within the data. Learning process management then prepares the model for training by selecting the appropriate algorithm for training, as well as the corresponding functions required for performance maintenance, while risk-checking the frameworks being used in the training environment. Model training merges the data management and process management steps to run the algorithm after which the data is validated using the validation dataset to evaluate the model's bias, variance, and quality of execution. Learning process verification uses the test dataset only. It evaluates the model's quality of execution, data bias, and data variance. It is not related in any way to validation, which is the last step of the model training stage. Model implementation moves the training model into one that may be run on the hardware targeted for the use case intended and any optimisations necessary are made in this stage in terms of computing requirements and necessities accommodated for. Inference model verification is the process in which the performance of the final inference model is evaluated through comparisons with the trained model. Additionally, compliance measures about software verifications are implemented according to ED-12C and DO-178C documentation.

TABLE 2 A use case advised by AIR6988. Shown is a predictive maintenance-involved system, where the ML-based system's functionality is summarized in the *Example* column, the *ID* column is "a unique identifier useful for reference in future work of the joint EUROCAE SAE G-34/WG-114 committee", the *Goal* details the ML-based system's functional operation is, *Inputs* counts the system's sensors and type of data, *Outputs* returns the message displayed as a result of the interaction between the ML-based system and the systems beneath, *Details* demonstrates the problems the use case targets, and *Integration* narrows down the system to be used with this AI application, whereas *Safety Concerns* raise severity level of the issues to be avoided for the completion of this use-case scenario.

Example	ID	Goal	Inputs	Outputs
Off-Board Predictive Maintenance	UC-SC322	Predict with high-specificity and high-accuracy an on-board failure with enough lead time to plan an optimized reaction	Low-level time-series sensor data collected and sent through a digital acquisition unit or data gateway	Failure message (can be EICAS/ECAMS message) + anticipated failure time + confidence of failure prediction
	Details		**Integration**	
	Combination of existing data cleansing/ETL + ML and other statistical methods to do big-data predictive maintenance		Aircraft owner, maintenance operation	
			Safety Concerns	
			Minimal, assuming existing procedures + instructions for parts handling are followed, and that scheduled maintenance is performed, as required	
Example	**ID**	**Goal**	**Inputs**	**Outputs**
On-Board Predictive Maintenance	UC-SC23	Predict with high-specificity and high-accuracy an on-board failure without having to send data to an off-board data center for analysis	Low-level time-series sensor data managed through high-bandwidth digital acquisition unit	EICAS/ECAMS message with predictive notation + anticipated failure time + confidence of failure prediction
	Details		**Integration**	
	Embedded NNs + other existing statistical methods (embedded) + on-board hardware for complex analytical processing		Aircraft owner, maintenance operation	
			Safety Concerns	
			Minimal, assuming existing procedures + instructions for parts handling are followed, and that scheduled maintenance is performed, as required	

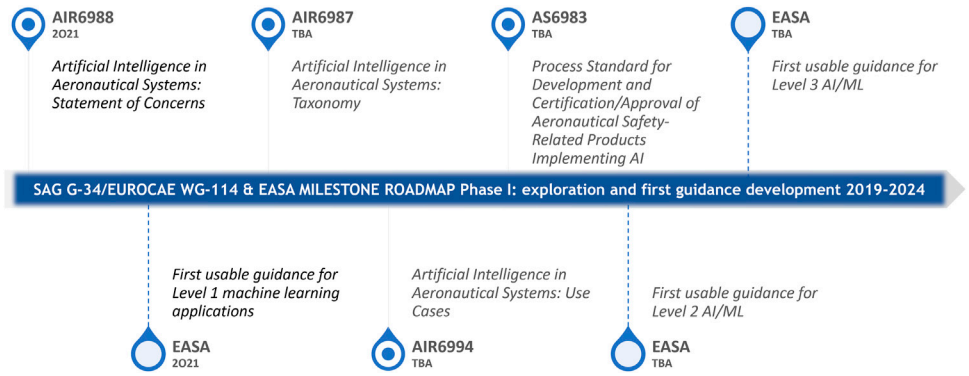


FIGURE 6
The standards for ML usage in aerospace are still being introduced, with a planned completion of frameworks by 2028 set by EASA. The first guidance development phase is to end by 2024, with the planned publication of ML use cases by the SAE workgroup, and the level 3 ML guidance paper by EASA. The SAE AIR6988 and EASA Level 1 ML Application guidance papers shown to the left of the timeline have been discussed in this paper.

(EASA and Daedalean AG, 2020) placed a layout for such a development life-cycle in the case of ML implementation, shown in Figure 5.

The process begins and ends with requirements management and verification while taking into reference ED-79A and ARP4754A documentation. In the midst of this life-cycle is data management where training, validation, and test datasets are collected and labelled as well as validated in comparison with the system requirements while sustaining a reliable amount of bias and variance within the data. Learning process management then prepares the model for training *via* selecting the appropriate algorithm for training as well as the corresponding functions required for performance maintenance, while risk-checking the frameworks being used in the training environment. Model training merges the data management and process management steps to run the algorithm after which the data is validated using the validation dataset in order to evaluate the model's bias, variance, and quality of execution. Learning process verification uses the test dataset only. It evaluates the model's quality of execution, data bias, and data variance. It is not related in any way to validation, which is the last step of the model training stage. Model implementation moves the training model into one that may be run on the hardware targeted for the use case intended and any optimisations necessary are made in this stage in terms of computing requirements and necessities accommodated for. Inference model verification is the process in which the performance of the final inference model is evaluated through comparisons with the trained model. Additionally, compliance measures with regard to software verifications are implemented according to ED-12C and DO-178C documentation (EASA and Daedalean AG, 2020).

The methodologies of certification discussed earlier may lead to a suggested use case advised by AIR6988 (SAE International, 2021a). The use case in Table 2 is an example of a predictive maintenance-involved system, where the ML-based system's functionality is summarized in the Example column, the ID column is "a unique identifier useful for reference in future work of the joint EUROCAE SAE G-34/WG-114 committee", the Goal details the ML-based system's functional operation is, Inputs counts the system's sensors and type of data, Outputs returns the message displayed as a result of the interaction between the ML-based system and the systems beneath, Details demonstrates the problems the use case targets, and Integration narrows down the system to be used with this AI application, whereas Safety Concerns raise severity level of the issues to be avoided for the completion of this use-case scenario.

7 A Roadmap and further research

Additional methods of data extraction for the use of ML, such as transfer learning, are currently being developed and seem promising for the benefit of this paper's direction. Transfer Learning is based on the development of a model's information for the use in another model performing similar tasks, while maintaining a low consumption of computationally-hungry processes and large amounts of data-requiring techniques. The aim is to keep the output and the task constant while changing the probability distributions required for the operation that leads to these tasks and outputs (EASA and Daedalean AG, 2020). Risks that may arise in correlation with resorting to such an approach include the necessity to verify the results of an empirical method-styled process, since transfer learning does include this approach. Another risk appears due to the requirement of transfer learning for a

"representative test set for the target function" (EASA and Daedalean AG, 2020), as a result of the source and target domain not being adequately related, causing an extra step and risk mitigation, trying to prevent what is known as a "negative transfer". Additional risk is due to uncertainty from using public dataset trained models as it may be more difficult to confirm that they comply with the learning assurance requirements. (Gardner et al., 2020) is an example of work in progress in this field, where the focus is on structures that have no data on their damage state obtained yet. The group uses data procured from an analogous structure to inference the damage on the former structure mentioned, using ML and non-destructive evaluation. The standards for ML usage in aerospace are still being introduced, with a planned completion of frameworks by 2028 set by EASA. Shown in Figure 6, the first guidance development phase is to end by 2024, with the planned publication of use cases by the SAE workgroup, and the level 3 ML guidance paper by EASA.

Author contributions

SP conceived the original idea and it was discussed with HM, whereby the main focus and paper's ideas were agreed upon with guidance taken from SP. The text of the paper is written by HM with sources for explainability and AI taken from colleagues, including those under the supervision of SP.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Alain, G., and Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. Available at: http://arxiv.org/abs/1610.01644 (Accessed January 6, 2022).

Au, J., Reid, D., and Bill, A. (2022). "Challenges and opportunities of computer vision applications in aircraft landing gear," in 2022 IEEE Aerospace Conference (Big Sky, MT), March 5–12, 2022. doi:10.1109/aero53065.2022.9843684

Azodi, C. B., Tang, J., and Shiu, S. H. (2020). Opening the black box: Interpretable machine learning for geneticists. *Trends Genet.* 36, 442–455. doi:10.1016/j.tig.2020.03.005

Ball, D. L., Norwood, D. S., and TerMaath, S. C. (2006). "Joint strike fighter airframe durability and damage tolerance certification," in 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 01 May 2006–04 May 2006 (Newport, Rhode Island. doi:10.2514/6.2006-1867

Barandela, R., Valdovinos, R. M., Salvador Sánchez, J., and Ferri, F. J. (2004). "The imbalanced training sample problem: Under or over sampling?," in. *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition* (Lisbon, Portugal: SSPR), 806. doi:10.1007/978-3-540-27868-9_88

Barter, S. A., Athiniotis, N., and Clark, G. (1992). "Cracking in an aircraft nose landing gear strut," in *Handbook of case histories in failure analysis*. Editor K. A. Esaklul (Ohio: ASM International), 11.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.

Blasch, E., Sung, J., Nguyen, T., Daniel, C. P., and Mason, A. P. (2019). "Artificial intelligence strategies for national security and safety standards," in *Aaai FSS-19: Artificial intelligence in government and public sector* (Arlington, Virginia. arXiv). doi:10.48550/arXiv.1911.05727

Cross, E., Sartor, P., Worden, K., and Southern, P. (2013). "Prediction of landing gear loads from flight test data using Gaussian process regression," in *Structural health monitoring 2013: A Roadmap to intelligent structures*. Editor F.-K. Chang (Lancaster, Pennsylvania: DEStech Publications), 1452

Cross, E., Sartor, P., Worden, K., and Southern, P. (2012). "Prediction of landing gear loads using machine learning techniques," in 6th European Workshop on Structural Health Monitoring (Dresden, Germany), July 3–6, 2012. Available at: http://www.ndt.net/?id=14124 (Accessed April 24, 2021).

EASA and Daedalean (2021). Concepts of design assurance for neural networks (CoDANN) II public extract. Available at: https://www.easa.europa.eu/en/document-library/general-publications/concepts-design-assurance-neural-networks-codann-ii (Accessed June 16, 2021).

EASA and Daedalean (2020). Concepts of design assurance for neural networks (CoDANN) public extract. Available at: https://www.easa.europa.eu/en/document-library/general-publications/concepts-design-assurance-neural-networks-codann (Accessed May 13, 2021).

DEEL Certification Workgroup (2021). White paper: Machine learning in certified systems (S079l03t00-005). Available at: https://hal.archives-ouvertes.fr/hal-03176080 (Accessed July 8, 2021).

Denney, E., Pai, G., and Pohl, J. (2012). "AdvoCATE: An assurance case automation toolset," in *Safecomp 2012: Computer safety, reliability, and security* (Magdeburg, Germany). 8–21. doi:10.1007/978-3-642-33675-1_2

Denney, E., and Trac, S. (2008). "A software safety certification tool for automatically generated guidance, navigation and control code," in 2008 IEEE Aerospace Conference (Big Sky, MT), 1–8 March 2008, Big Sky, Montana. doi:10.1109/AERO.2008.4526576

Dodge, J., Vera Liao, Q., Zhang, Y., Bellamy, R. K. E., and Dugan, C. (2019). "Explaining models: An empirical study of how explanations impact fairness judgment," in IUI '19: 24th International Conference on Intelligent User Interfaces, March 16–20, 2019, Marina del Ray, California (California), 275–285. doi:10.1145/3301275.3302310

Dziendzikowski, M., Kurnyta, A., Reymer, P., Kurdelski, M., Klysz, S., Leski, A., et al. (2021). Application of operational load monitoring system for fatigue estimation of main landing gear attachment frame of an aircraft. *Materials* 14, 6564. doi:10.3390/ma14216564

EASA (2020). Artificial intelligence Roadmap: A human-centric approach to AI in aviation. Available at: https://www.easa.europa.eu/en/newsroom-and-events/news/easa-artificial-intelligence-roadmap-10-published (Accessed May 13, 2021).

EASA (2021). EASA Concept paper: First usable guidance for level 1 machine learning applications: A deliverable of the EASA AI Roadmap. Available at: https://www.easa.europa.eu/en/newsroom-and-events/news/easa-releases-its-concept-paper-first-usable-guidance-level-1-machine-0 (Accessed February 22, 2022).

EUROCAE (2021). ER-022: Artificial intelligence in aeronautical systems: Statement of Concerns. Available at: https://eshop.eurocae.net/eurocae-documents-and-reports/er-022/# (Accessed December 8, 2021).

Fatemi, A., and Vangt, L. (1998). Cumulative fatigue damage and life prediction theories: A survey of the state of the art for homogeneous materials. *Int. J. Fatigue* 20, 9–34. doi:10.1016/S0142-1123(97)00081-9

Federal Aviation Administration (2005). AC 23-13a: Fatigue, fail-safe, and damage tolerance evaluation of metallic structure for normal, utility, acrobatic, and commuter category Airplanes. Available at: https://www.faa.gov/regulations_policies/advisory_circulars/index.cfm/go/document.information/documentid/22090 (Accessed October 4, 2020).

Fulton, R., and Vandermolen, R. (2017). *Airborne electronic hardware design assurance*. Boca Raton, FL: CRC Press.

Gardner, P., Fuentes, R., Dervilis, N., Mineo, C., Pierce, S. G., Cross, E. J., et al. (2020). Machine learning at the interface of structural health monitoring and non-destructive evaluation. *Phil. Trans. R. Soc. A* 378, 20190581. doi:10.1098/rsta.2019.0581

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.* 51, 1–42. doi:10.1145/3236009

Guyon, I., and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, 1157

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006). *Feature extraction: Foundations and applications*. Heidelberg: Springer-Verlag.

Hall, P., Ambati, S., and Phan, W. (2017). *Ideas on interpreting machine learning*. O'Reilly Media. Available at: https://www.oreilly.com/radar/ideas-on-interpreting-machine-learning/ (Accessed September 25, 2021).

Hawkins, R., Paterson, C., Picardi, C., Jia, Y., Calinescu, R., and Habli, I. (2021). Guidance on the assurance of machine learning in autonomous systems (AMLAS). Available at: https://www.york.ac.uk/assuring-autonomy/guidance/amlas/ (Accessed May 13, 2021).

Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). "Explaining collaborative filtering recommendations," in *CSCW00: Computer supported cooperative work*. (Pennsylvania, Philadelphi. doi:10.1145/358916.358995

Holmes, G., Sartor, P., Reed, S., Southern, P., Worden, K., and Cross, E. (2016). Prediction of landing gear loads using machine learning techniques. *Struct. Health Monit.* 15, 568–582. doi:10.1177/1475921716651809

Hoole, J. G. (2020). *Probabilistic fatigue methodology for aircraft landing gear*. Ph.D. Thesis. University of Bristol. Available at: https://hdl.handle.net/1983/8061165f-0a39-4532-bbbd-029e99286706 (Accessed June 3, 2021).

Hunt, S. R., and Hebden, I. G. (2001). Validation of the Eurofighter Typhoon structural health and usage monitoring system. *Smart Mat. Struct.* 10, 497–503. doi:10.1088/0964-1726/10/3/311

International Organization for Standardization (2022). ISO/PAS 21448:2022: Road vehicles-safety of the intended functionality. Available at: https://www.iso.org/standard/77490.html (Accessed August 20, 2022).

Irving, P. E., Strutt, J. E., Hudson, R. A., Allsop, K., and Strathern, M. (1999). The contribution of fatigue usage monitoring systems to life extension in safe life and damage tolerant designs. *Aeronaut. J.* 103, 589–595. doi:10.1017/S0001924000064228

Kaakai, F., Dmitriev, K., Adibhatla, S., Shreeder")Baskaya, E., Bezzecchi, E., et al. (2022). Toward a machine learning development lifecycle for product certification and approval in aviation. *SAE Int. J. Aerosp.* 15, 9. doi:10.4271/01-15-02-0009"

Kaur, H., Pannu, H. S., and Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.* 52, 1–36. doi:10.1145/3343440

Khan, S., Phillips, P., Hockley, C., and Jennions, I. (2014a). No Fault Found events in maintenance engineering Part 2: Root causes, technical developments and future research. *Reliab. Eng. Syst. Saf.* 123, 196–208. doi:10.1016/j.ress.2013.10.013

Khan, S., Phillips, P., Jennions, I., and Hockley, C. (2014b). No Fault Found events in maintenance engineering Part 1: Current trends, implications and organizational practices. *Reliab. Eng. Syst. Saf.* 123, 183–195. doi:10.1016/j.ress.2013.11.003

Khan, S., and Yairi, T. (2018). A review on the application of deep learning in system health management. *Mech. Syst. Signal Process.* 107, 241–265. doi:10.1016/j.ymssp.2017.11.024

Kocielnik, R., Amershi, S., and Bennett, P. N. (2019). "Will you accept an imperfect AI? Exploring designs for adjusting end-user expectations of AI systems," in CHI '19: CHI Conference on Human Factors in Computing Systems, May 4–9, 2019 (Glasgow, Scotland. doi:10.1145/3290605.3300641)

Kursa, M. B., and Rudnicki, W. R. (2011). The all relevant feature selection using random forest. Available at: http://arxiv.org/abs/1106.5112 (Accessed January 7, 2022).

Ladda, V., and Struck, H. (1991). Operational loads on landing gear." in 71st Meeting of the AGARD Structures and Materials Panel (Povoa de Varzim, Portugal), 8th–12th October 1990. Available at: https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/N9128158.xhtml (Accessed November 18, 2020).

Le-The, Q.-V. (2016). *Application of multiaxial fatigue analysis methodologies for the improvement of the life prediction of landing gear fuse pins*. MSc Thesis. Carleton University. doi:10.22215/etd/2016-11572

Liu, Y., Loh, H. T., and Sun, A. (2009). Imbalanced text classification: A term weighting approach. *Expert Syst. Appl.* 36, 690–701. doi:10.1016/j.eswa.2007.10.042

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill.

Montavon, G., Samek, W., and Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* 73, 1–15. doi:10.1016/j.dsp.2017.10.011

Murphy, C., Kaiser, G., and Arias, M. (2006). *A framework for quality assurance of machine learning applications*. CUCS-034-06. doi:10.7916/D8MP5B4B

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., et al. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artif. Intell. Rev.* 52, 77–124. doi:10.1007/s10462-018-09679-z

Pascual, F. G., and Meeker, W. Q. (1999). Estimating fatigue curves with the random fatigue-limit model. *Technometrics* 41, 277–289. doi:10.1080/00401706.1999.10485925

Poggio, T., Banburski, A., and Liao, Q. (2020). Theoretical issues in deep networks. *Proc. Natl. Acad. Sci. U. S. A.* 117, 30039–30045. doi:10.1073/pnas.1907369117

Pu, P., and Chen, L. (2006). "Trust building with explanation interfaces," in IUI06: 11th International Conference on Intelligent User Interfaces, 29 January 2006–1 February 2006 (Sydney, Australia). doi:10.1145/1111449.1111475

RTCA (2012). *DO-178C software considerations in airborne systems and equipment certification*. Washington, DC: RTCA, Inc.

RTCA (2000). *DO-254: Design assurance guidance for airborne electronic hardware*. Washington, DC: RTCA, Inc.

Russell, S., and Norvig, P. (2022). *Artificial intelligence: A modern approach*. Global Edition 4th ed. Harlow, UK: Pearson.

Sae International (2021a). AIR6988: Artificial intelligence in aeronautical systems: Statement of Concerns. Available at: https://saemobilus.sae.org/content/AIR6988 (Accessed May 13, 2021).

Sae International (2010). ARP4754A: Guidelines for development of civil aircraft and systems. Available at: https://www.sae.org/standards/content/arp4754a (Accessed May 6, 2020).

Sae International (2021b). ARP6461A: Guidelines for implementation of structural health monitoring on fixed wing aircraft. Available at: https://saemobilus.sae.org/content/ARP6461A (Accessed February 3, 2022).

Schmidt, R. K. (2021). "*The design of aircraft landing gear*," (Warrendale, PA: SAE International), 858

Schwartz-Ziv, R., and Tishby, N. (2017). Opening the black box of deep neural networks via information. Available at: https://arxiv.org/abs/1703.00810 (Accessed February 28, 2022).

Smith-Renner, A., Fan, R., Birchfield, M., Wu, T., Boyd-Graber, J., Weld, D. S., et al. (2020). "No explainability without accountability: An empirical study of explanations and feedback in interactive ML," in CHI '20: CHI Conference on Human Factors in Computing Systems (Honolulu, USA, 1–13. doi:10.1145/3313831.3376624

Staszewski, W. J., and Boller, Chr. (2004). "Aircraft structural health and usage monitoring," in *Health monitoring of aerospace structures: Smart sensor technologies and signal processing* (Chichester: J. Wiley), 29.

Sweeney, D. L. (2015). "Understanding the role of RTCA DO-160 in the avionics certification process," in *Digital avionics handbook* (Boca Raton: Taylor &Francis Group), 194.

Tao, J. X., Smith, S., and Duff, A. (2009). The effect of overloading sequences on landing gear fatigue damage. *Int. J. Fatigue* 31, 1837–1847. doi:10.1016/j.ijfatigue.2009.03.012

Ten Zeldam, S. (2018). *Automated failure diagnosis in aviation maintenance using eXplainable artificial intelligence (XAI)*. MSc Thesis. University of Twente. Available at: https://purl.utwente.nl/essays/75381 (Accessed March 11, 2022).

Viaña, J., Ralescu, S., Ralescu, A., Cohen, K., and Kreinovich, V. (2022). Explainable fuzzy cluster-based regression algorithm with gradient descent learning. *Complex Eng. Syst.* 2, 8. doi:10.20517/ces.2022.14

Wanhill, R. J. H. (2018). "Fatigue requirements for aircraft structures," in *Aircraft sustainment and repair*, Editor R. Jones, A. Baker, N. Matthews, and V. Champagne (Oxford: Elsevier), 17–40. doi:10.1016/b978-0-08-100540-8.00002-9

Wong, J., Ryan, L., and Kim, I. Y. (2018). Design optimization of aircraft landing gear assembly under dynamic loading. *Struct. Multidiscipl. Optim.* 57, 1357–1375. doi:10.1007/s00158-017-1817-y

Xu, Z., Shen, D., Nie, T., and Kou, Y. (2020). A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data. *J. Biomed. Inf. X.* 107, 103465. doi:10.1016/j.jbi.2020.103465

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. Available at: http://arxiv.org/abs/1611.03530 (Accessed January 11, 2022).



https://dspace.org/

Certification of machine learning algorithms for safe life assessment of landing gear

El Mir, Haroun; Perinpanayagam, Suresh

2022-11-15
Frontiers
https://doi.org/10.3389/fspas.2022.896877
El Mir H & Perinpanayagam S (2022) Certification of machine learning algorithms for safe life assessment of landing gear, Frontiers in Astronomy and Space Sciences, Volume 9, November 2022, Article number 896877

Downloaded from a DSpace repository. https://dspace.org/