



HAL
open science

On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components

Florence de Grancey, Sébastien Gerchinovitz, Lucian Alecu, Hugues Bonnin,
Joseba Dalmau, Kevin Delmas, Franck Mamalet

► **To cite this version:**

Florence de Grancey, Sébastien Gerchinovitz, Lucian Alecu, Hugues Bonnin, Joseba Dalmau, et al..
On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components. ERTS
2024, Jun 2024, Toulouse, France. hal-04575318v2

HAL Id: hal-04575318

<https://hal.science/hal-04575318v2>

Submitted on 28 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Feasibility of EASA Learning Assurance Objectives for Machine Learning Components

Florence de Grancey, Thalès Avionics
Sébastien Gerchinovitz, IRT Saint Exupéry and IMT
Lucian Alecu, Continental
Hugues Bonnin, Continental
Joseba Dalmau, IRT Saint Exupéry
Kevin Delmas, ONERA
Franck Mamalet, IRT Saint Exupéry

Florence.de-Grancey@fr.thalesgroup.com
Sebastien.Gerchinovitz@irt-saintexupery.com
Lucian.Alecu@continental-corporation.com
Hugues.Bonnin@continental-corporation.com
Joseba.Dalmau@irt-saintexupery.com
Kevin.Delmas@onera.fr
Franck.Mamalet@irt-saintexupery.com

Abstract— Despite the significant success of using Machine Learning (ML) in numerous industrial applications, how to integrate these technologies in safety-critical contexts poses many challenging questions. Several industrial and academic research groups, as well as various standardization committees are actively working to provide (partial) answers to these questions. In this document, we focus on one such initiative led by the EASA, which proposes a series of guidelines and requirements to develop ML-based systems for critical applications in the aviation domain. In this paper we investigate whether these requirements can be satisfied when using ML to solve a relatively simple regression task, that of building a neural network surrogate of the International Geomagnetic Reference Field (IGRF) model. Though we acknowledge all the structuring efforts towards the ambitious certification goal, our analysis pinpoints several important issues with some of these guidelines, such as ambiguous definitions, prohibitive computational costs, or currently very limited theoretical guarantees. Our analysis compels us to remain cautious about the various general recommendations proposed for designing trustworthy ML components for safety-critical systems. These conclusions call for the academic and industrial communities concerned by "Trustworthy AI" to strengthen their collaboration and pursue the research efforts necessary to address the existing challenges and establish sound methodologies for building safe ML-based applications.

Keywords— machine learning, safety, guidelines, certification, trustworthiness.

I. CONTEXT

In recent years, we have witnessed a multitude of ongoing initiatives to establish recommendations, guidelines and norms on how to develop and certify trustworthy Machine Learning (ML) solutions for safety-critical systems in the context of several application domains. One such initiative in the aviation domain is led by the European Union Aviation Safety Agency (EASA). In early 2023 the EASA released an open version of the "EASA concept paper: first usable guidance for level 1&2 machine learning applications", updated in March 2024 [1]. The document proposes a series of guidelines aimed at increasing the trustworthiness of ML components intended for aviation-related safety-critical applications. The authors formulate several objectives which, in their view, must be met to certify such technologies.

II. CONCEPT PAPER OVERVIEW

Ensuring that a data-driven software component is trustworthy raises numerous challenges. The EASA concept paper attempts to provide a holistic design methodology for ML-based systems in the aviation domain. In this section we briefly describe the

structure of the concept paper and point out the requirements we choose to analyze.

The EASA concept paper is structured around four main blocks: *AI Trustworthiness analysis*, *AI assurance*, *Human factors for AI* and *AI safety risk mitigation*. The safety assessment lies at the heart of the first block. It is within this phase that a system is assigned its main objectives in terms of safety, in particular, the assessment of the impact of a system failure on its environment (and notably on human lives), i.e. the dangerousness of the failures. The other blocks complete this assessment from different angles. *AI assurance* reinforces the level of trust in the AI system itself: on the one hand via "learning assurances" that "cover the paradigm shift from programming to learning", on the other hand via "development explainability", which seeks to open the "black box" that is machine learning. The remaining two blocks participate in safety "from the outside" of the system: the *Human factors for AI* cover the aspects of the relationship of the system with its user/operator, while the *AI safety risk mitigation* covers the residual risks identified by the *AI Trustworthiness analysis*.

We focus on AI/ML component safety only, because our field of research focuses on the ML models themselves. We seek to evaluate both the intrinsic risks of ML components, as well as the means of mitigation of these risks, which are also directly associated to the models. In this context, our analysis will focus on the objectives of the blocks *AI Trustworthiness analysis* (SA) and *AI assurance* (LM) only.

LM objectives can be divided into two categories: the objectives pertaining to the transparency and consistency of the engineering process, and the objectives related to the exploitation of quantitative and mathematical elements of the AI/ML models.

We do not address the objectives related to the engineering process, because they are classical and relatively indisputable. These objectives mainly request that each of the engineering activities must be clearly defined, traced and verified. In this set of objectives, the causal relationship between the measures taken and the safety risk is obvious, since it ensures that there is no discrepancy (or that it is as minimal as possible) between the discourse and the reality of engineering. Indeed, a lack of transparency and consistency in the engineering process undermines the whole safety demonstration.

The LM objectives related to the quantitative and mathematical elements of the ML model (the ones we focus on in this paper) are the following:

- LM-04: Quantifiable generalisation bounds

- LM-07: Bias-variance trade-off
- LM-08: Bias-variance requirement
- LM-09: Performance result
- LM-11: Stability analysis of the learning algorithm
- LM-12: Stability of the trained model
- LM-13: Model robustness
- LM-14: Verification of the anticipated generalisation bounds

Let us highlight that the EASA concept paper establishes a strong link between some objectives of the SA and of the LM. This link is implemented in the objectives relating to performance (LM 09), generalization (LM 04) and safety assessments (SA 01). Indeed, the anticipated Mean of Compliance of SA 01 objective indicates that "as part of the safety assessment process, AI/ML item failure modes are expected to be identified. Performance metrics should provide a conservative estimation of the probability of occurrence of the AI/ML item failures modes". The LM 09 and IMP 09 objectives are then referenced in the same paragraph as participating in this estimation, in connection with the LM04 generalization objective, which allows pronouncing on the failure rate in operation. Therefore, even if we do not analyze SA objectives directly, we discuss in Section V the link between LM objectives and safety.

The goal of the present paper is to evaluate the feasibility of the above LM objectives. However, doing so for a new operational use case is notably hard, mainly due to the cost of data acquisition. Therefore, we choose to focus on the magnetic declination estimation use case, a surrogate modelling problem (cf. the technical details in the next section). We have chosen this particular use-case for three main reasons:

- it is suitable for integration into an airborne system,
- a ML-based approach appear promising as compared to more traditional approaches,
- both data and algorithms are readily available.

For most common ML tasks, the ground truth values are either unknown, or observed via a noisy measurement process. The case of surrogate modelling is simpler, since it aims at approximating existing complex functions with ML models, and the ground truth values are therefore known. As such, the LM objectives are easier to evaluate for our surrogate modelling use case than for other ML tasks. We thus anticipate that the challenges identified in this work about the application of the LM objectives will also hold for other (more complex) use cases.

For the magnetic declination estimation use case, we can derive system/ML requirements using the requirements on magnetic heading provided in [2], which are performance oriented. Ensuring these requirements is considered as sufficient to demonstrate trustworthiness. Consequently, while the proposed use case may not strictly fall under the EASA guidelines for critical airborne systems, it still presents a realistic, well-defined, and thoroughly documented system. Moreover, the study performed on this use case is mostly generalizable to other surrogate software items used in critical embedded systems. In the upcoming sections, along with the analysis of our surrogate modelling use case, we also discuss the generalization of our findings to other types of ML tasks.

III. USE CASE AND APPROACH

In this section, we describe the magnetic declination estimation use case in detail, as well as our analysis approach, including the experiment setup.

Use case. We consider the following use case: build a neural network surrogate model of the International Geomagnetic Reference Field (IGRF) produced by IAGA. The IGRF describes the Earth's main magnetic field, by modelling the geomagnetic potential as a finite series of spherical harmonics. The latest generation, IGRF-13 [3], involves Schmidt semi-normalized associated Legendre functions of degree up to $n=13$, and provides the values of all spherical harmonics Gauss coefficients (which vary over time) at various 5-year-spaced epochs. IGRF-13 enables users to compute the magnetic field components in three dimensions, the magnetic inclination and the magnetic declination at each location on and above the Earth's surface, from 1900 to the present.

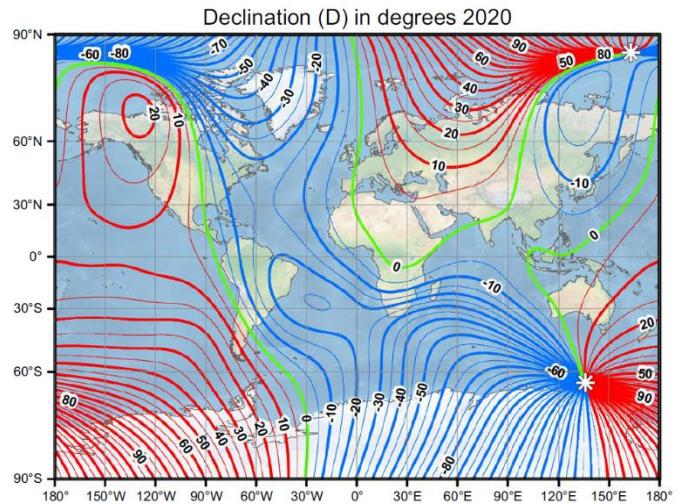


Figure 1: Declination map at the WGS84 ellipsoid surface for epoch 2020 (source: Alken et al. [1], Fig 1).

In the sequel we focus on the magnetic declination (also known as magnetic variation), which is the angle between the true North and the magnetic North. The magnetic declination depends on the latitude, longitude, altitude, and time; see Figure 1 for an illustration. The magnetic declination model is currently embedded in large commercial aircrafts to compute the aircraft's magnetic heading in real time. The magnetic heading data is crucial during the landing process, as airport diagrams still describe runways by their magnetic heading. However, for some aeronautics systems, embedding limitations prevent the use of the complete IGRF model and call for using a computationally more tractable model. To that end, and for illustrative purposes, we approximate the magnetic declination computation done by the IGRF-13 with a shallow neural network surrogate (see the experiment setup details below).

We stress that our goal is not to produce the most efficient or accurate surrogate model, but rather to propose a simple real-world use case on which the LM objectives can be instantiated. However, to keep things realistic, we consider the performance requirements on magnetic heading provided in [2].

Latitude range	Acceptable accuracy (95%)
50°S - 50°N	2°
50°N - 73°N	3°
60°S - 50°S	3°
73°N - 79°N	5°
79°N - 82°N	8°

Table 1: Acceptable accuracy values for magnetic heading [2]

Approach. We analyse the LM objectives of the EASA concept paper listed in Section II. For each of the objectives, we study the following (related) aspects:

- *Clarity*: is the objective clearly formulated or might it be prone to ambiguous interpretations?
- *Applicability*: does the objective apply to the considered use case?
- *Feasibility*: can the objective be achieved within reasonable costs and in a timely manner? Can it be formally and/or empirically assessed?

For each of the objectives, we start our analysis by instantiating generic definitions and principles to our use case. Then, we seek and apply well-established methods and results in the scientific literature to fulfill these requirements. Our experiment setup is described in the paragraph below. We also identify the hypotheses that must be met to ensure the validity of these approaches. Finally, we assess their practical feasibility and computational complexity.

Experiment setup. We use the python tool PyIGRF as the ground truth reference [4]. We restrict our study to the latitude range 60°S-82°N for which performance requirements are available (see Table 1). We consider all locations within that range, at an altitude of 100 meters and for the year 2005 for simplicity. This defines the Operational Design Domain (ODD).

We build three independent datasets that will prove useful in the next sections. The letters θ and ϕ denote the latitude and longitude.

1. Training set: it consists of 750K points $x_i = (\theta_i, \phi_i)$ drawn independently at random, uniformly within the latitude range 60°S-82°N and longitude range 180°W-180°E. This dataset is used for model training, that is, to build the neural network surrogate.
2. Calibration set: it consists of 10K points (θ_i, ϕ_i) drawn independently at random, uniformly within the latitude range 60°S-82°N and longitude range 180°W-180°E. This dataset is used in order to obtain estimates about the trained model, for the objectives pertaining to generalisation (see Section IV.F).
3. Test set: it consists of 250K points (θ_i, ϕ_i) drawn independently at random, uniformly within the latitude range 60°S-82°N and longitude range 180°W-180°E. This dataset is used only for test purposes.

Our surrogate model is a neural network having the following architecture: a fully connected ReLU neural network with 3 hidden layers, and 20 neurons per layer. We provide four scalar inputs to the neural network: $\cos(\theta)$, $\sin(\theta)$, $\cos(\phi)$, $\sin(\phi)$. The network has one scalar output modelling the magnetic declination. The outputs are normalized to [0,1]. We train the network to fit the magnetic declination (obtained with PyIGRF) on the training set, using the square loss with the SGD optimizer, a learning rate of 0.005, batches of size 32 and 15 epochs. We thus obtain a *trained model*.

Notation. We denote by f the true IGRF-13 model. We denote by S the training set and by \hat{f}_S the trained surrogate model built as explained in the paragraph above and trained used the training set S .

IV. RESULTS

In this section, we instantiate all the aforementioned LM objectives to this specific use case, and we evaluate them in terms of clarity, applicability, and feasibility.

A. Analysis of objective LM-09: Performance on test set

The first objective is about the performance of the trained model.

Objective LM-09: The applicant should perform an evaluation of the performance of the trained model based on the test data set and document the result of the model verification.

To achieve this objective we evaluate appropriate metrics over a “representative” test set. There is no particular issue concerning the *clarity*, *applicability* and *feasibility* of this verification step. For the IGRF surrogate model, representativity is simple since we can build the test data set as desired. Moreover, the performance metric is defined as the 95% quantile of all absolute errors on the test set, where an *absolute error* (also termed *accuracy* thereafter) is the absolute difference between the true and predicted magnetic declination values. Results are displayed in Table 2.

Latitude range	95% accuracy on test set
50°S - 50°N	1.51°
50°N - 73°N	1.97°
60°S - 50°S	3.14°
73°N - 79°N	3.47°
79°N - 82°N	5.00°

Table 2: Accuracy of the trained model when evaluated on the test set. We report the 95% empirical quantiles of the absolute errors (accuracies) on each latitude range.

Note that the trained model seems accurate enough in that the 95% accuracies on the test set almost meet the performance requirements of Table 1.

Even if the LM-09 objective is feasible for this surrogate use case, the choice of the adequate performance metrics may be a complex activity for the applicant. Notably more so if the applicant is dealing with computer vision or natural language processing models, where common metrics have a less clear-cut interpretation.

B. Analysis of objectives LM-07 and LM-08: Bias-Variance

The next objectives are about the Bias-Variance trade-off.

Objective LM-07-SL: The applicant should account for the bias-variance trade-off in the model family selection and should provide evidence of the reproducibility of the model training process.

Objective LM-08: The applicant should ensure that the estimated bias and variance of the selected model meet the associated learning process management requirements.

These two objectives seem, at first glance, both justifiable and achievable. Informally speaking, achieving a low bias and low variance corresponds to learning a sufficiently expressive model that does not depend too much on the training set. For many ML models, achieving low bias and low variance simultaneously should constitute a good indication of a well-performing predictive model. Despite these first intuitions, our analysis

shows that, even for the surrogate model case, the satisfaction of these objectives is not straightforward.

From a theoretical point of view, we can often consider, at least intuitively for regression tasks, the mean least square error decomposition into bias and variance. For a given example x , this decomposition expresses the expected squared error as the sum of a bias term (squared), a variance term, and a noise term:

$$\mathbb{E}_{S,y} [(\hat{f}_S(x) - y)^2] = \left(\mathbb{E}_S[\hat{f}_S(x)] - f(x) \right)^2 + \mathbb{E}_S [(\hat{f}_S(x) - \mathbb{E}_S[\hat{f}_S(x)])^2] + \mathbb{E}_y[(y - f(x))^2]$$

In the above equation, $\mathbb{E}_{S,y}$ means that we consider averages over all training sets S of a given size and all possible labels y for a fixed input x . The notation \mathbb{E}_S and \mathbb{E}_y are understood similarly. We can identify:

- the bias: $B(x) = \mathbb{E}_S[\hat{f}_S(x)] - f(x)$
- the variance: $V(x) = \mathbb{E}_S [(\hat{f}_S(x) - \mathbb{E}_S[\hat{f}_S(x)])^2]$
- the variance of the noise: $\sigma^2(x) = \mathbb{E}_y [(y - f(x))^2]$

In our surrogate use case, the variance of the noise equals zero. We perform a **rough estimation** of the bias and variance terms with a bootstrap method [5]. It consists in performing M experiments where a new data set $S(i)$ is drawn by sampling with replacement inside S . For the bias term, we estimate $\mathbb{E}_S[\hat{f}(x)]$ with $\frac{1}{M} \sum \hat{f}_{S(i)}(x)$ for each x , subtract the known value of $f(x)$, and average the squared result over all values of x in the test set. We proceed similarly for the variance. Results are shown in Figure 2, for a reduced training dataset of $n=25K$ points and $M=200$ bootstrap experiments. These estimates are repeated for several values of model complexity corresponding to the number of neurons per layer.

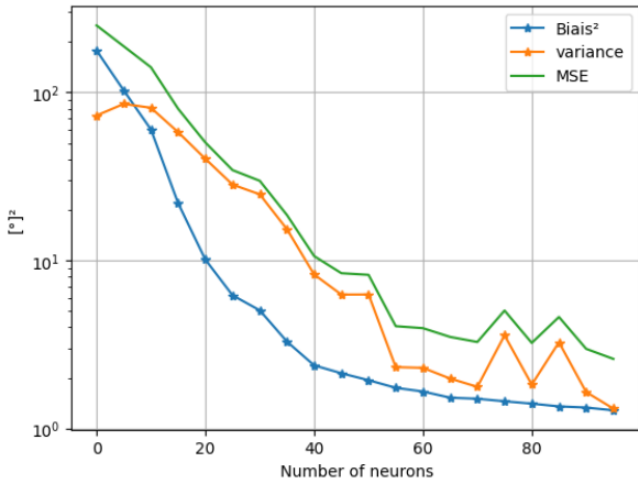


Figure 2: Rough estimation of MSE decomposition terms, for a variable number of neurons per layer.

Note that both the bias and variance terms are (roughly) decreasing for layer widths larger than 5. These rough observations are reminiscent of the double descent phenomenon in deep learning. This could lead the applicant to choose the largest network among those evaluated, while smaller networks (with about 40 neurons per layer in our case) might already be sufficiently accurate. This might raise embedding challenges. We thus argue that the bias-variance estimation may not always be the best tool to select an ML model architecture.

¹ Indeed, for an overparametrized neural network that can easily overfit the training data, the in-sample error can be zero, while the bias and the variance can be positive.

These considerations and experiments allow us to provide the following answers with respect to the criteria enumerated above:

- *clarity*: bias and variance are mathematical notions that are easily misinterpreted. Since the suggested informal definitions in the concept paper (see Anticipated MOC LM-08) are ambiguous and possibly different from the traditional notions¹, we used instead the formal definitions above, which are in line with those of the CoDANN report [6].
- *applicability*: these definitions should be specialized to the learning task at hand and the performance metrics used. Applying them to the absolute error metric (which would be more consistent with our use case) instead of the squared error metric is not straightforward. Applications to classification use cases would raise similar difficulties. A unified framework for bias-variance decomposition is proposed in [7, 8], but this decomposition is complex (the performances may not decompose as a sum of bias and variance terms) and not feasible in general.
- *feasibility*: While estimating the bias is possible in this surrogate model context, this is not the case for general ML problems, where the true value $f(x)$ is typically unknown. Furthermore, even in our setting, estimating the bias-variance tradeoff is computationally prohibitive as it requires training an important number of models (number of settings of complexity parameter, times number of bootstrap experiments).

Our analysis shows that attempting to satisfy this seemingly intuitive criterion for the trustworthiness of ML models can raise significant technical and methodological challenges. This calls for further academic research efforts. It would also be useful to investigate the quantitative link between an optimal bias-variance tradeoff and the resulting ML performances for several task-specific metrics.

C. Analysis of objective LM-11: Learning algorithm stability

The next selected objective is about stability of the learning algorithm.

Objective LM-11: The applicant should provide an analysis on the stability of the learning algorithms.

This objective aims at assessing the reproducibility of the learning process. As no anticipated means of compliance is provided in the concept paper, we choose to rely on the definition provided in [9]: Assume A is a symmetric learning algorithm², which given a training set $S = \{z_i = (x_i, y_i), i = 1, \dots, n\}$, outputs a function \hat{f}_S (a model) mapping x to y . For any i and any new sample $z' = (x', y')$, consider the modified training set $S_i = (S \setminus \{z_i\}) \cup \{z'\}$ obtained by replacing z_i with z' in S . The algorithm A is called β -stable if, for any training set S , any i , and any new sample z' , the losses of the models \hat{f}_S and \hat{f}_{S_i} on any sample $z = (x, y)$ differ by at most β . More formally, the algorithm A is called β -stable if

$$\forall S, \forall i, \forall z', \forall z, \quad |\text{loss}(\hat{f}_S, z) - \text{loss}(\hat{f}_{S_i}, z)| \leq \beta.$$

² To be rigorous, this symmetry assumption does not hold in our case (we use batch stochastic gradient descent). Though this assumption is useful for the theoretical guarantees proved in [9], the rest of the definition still makes sense without it.

This definition helps to define a process to assess learning stability:

- create new training datasets S_i by modifying one sample of the training dataset S and train a replacement model \hat{f}_{S_i} ;
- compute, for any sample z of the test dataset, the absolute value of the loss difference between the trained model \hat{f}_S and the replacement model \hat{f}_{S_i} ;
- find the maximal absolute difference, which should be lower than a given threshold β .

In our surrogate use case, we follow this process to empirically estimate a lower bound of β through Monte Carlo experiments on M modified datasets and their corresponding trained ML models. Due to the extensive computational cost, we only use a reduced initial training dataset $n=25K$, and $M=200$ modified training sets. For each experiment, we evaluate the maximum absolute difference over the test set. We also experiment with two different design choices to evaluate their influence on the estimated lower bounds: the first one uses the same weights initialization for all trainings, the second one uses independent random weights initialization for each experiment.³

Figure 3 presents the results obtained for each training iteration. We observe that the estimated β parameter is very high in the random weight initialization case. Even with a fixed weight initialization, the variation of the loss can be high which is difficult to interpret (optimization problem, parameters choice, complexity of the ground truth function to approximate,...).

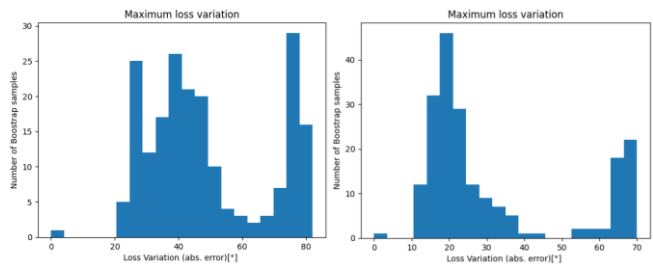


Figure 3 Maximum loss variation across the M training sets – (left) with random weights initialization – (right) with fixed weight initialization. Maximum value represents a lower bound on the β threshold of learning algorithm stability.

These considerations and experiments allow us to provide the following answers with respect to the criteria enumerated above:

- *clarity*: even if the formal definition given above seems understandable, it raises the challenge of the specification of the appropriate β parameter. The interpretation of this parameter represents the worst case over all training datasets S , all their modifications S_i , and all ODD points z .
- *applicability*: despite this favorable use case (surrogate modeling), we cannot find a sound choice of the threshold β . We must notice that usual ML learning processes rely on randomness (e.g., dataset shuffling, random ML model weights initialization) and on the choice of an optimizer. These elements all constitute a source of variability that can lead to the un-stability of performances in several points in the input space. Consequently, the design choices and the knowledge of the training framework highly affect the *applicability* of the learning process stability assessment.
- *feasibility*, considering that few or no formal methods are found in the literature even for our use case, we rely on a

Monte Carlo estimation of the parameter β . Such an evaluation is computationally expensive (for instance we had to work with a limited training set in our case), and would be even more challenging for large models and datasets. Moreover, performing this objective requires the evaluation of each source of variability in the learning process, which may be not be feasible with a black-box training framework. Furthermore, for general ML problems, a major difficulty may also come from the impossibility of generating new dataset samples.

To conclude, even for this surrogate use case, the interpretation of the evaluated β parameter is not clear, since the variation of the loss can depend on sources other than the dataset, such as the optimization process. Thus, we argue that the choice of the β parameter during model design is almost impossible, as it must bound all possible loss differences across the choice of the changed example.

D. Analysis of objective LM-12: Trained model stability

The next objectives are about the stability of the trained model \hat{f}_S .

Objective LM-12: The applicant should perform and document the verification of the stability of the trained model, covering the whole AI/ML constituent ODD.

The Anticipated MOC LM-12-1 gives only an informal definition as the evaluation of “perturbations in the operational phase due to fluctuations in the data input (e.g. noise on sensors) and having a possible effect on the trained model output”. We can rely on the formal definition given in [6]: given two thresholds δ and ϵ , stability is assessed by evaluating if :

$$\forall x, x' \in ODD, \quad \|x - x'\| \leq \delta \Rightarrow |\hat{f}_S(x) - \hat{f}_S(x')| \leq \epsilon.$$

The values δ and ϵ are supposed to be given in the ML component requirements, but the choice of δ and ϵ raises several challenges, as we point out both below and in Section V.B.

Clarity: At a first glance, this definition seems understandable and easy to achieve. However, even for a surrogate task it may not be adequate: if the ground truth function presents high local variations (large Lipschitz constant) in some parts of the ODD, a good surrogate ML model \hat{f}_S will also vary greatly. In particular, the surrogate model will only be able to fulfill the above condition for either very high values of ϵ , or for very small values of δ . Of course, such a choice of δ and ϵ is too conservative in regions of the ODD where the Lipschitz constant of the ground truth function is small, and does not at all guarantee that the surrogate model will be stable in such regions.

Applicability: For the IGRF use case, performance objectives are given in Table 1; we can therefore specify an acceptable ϵ threshold based on these performance requirements. For the δ threshold describing position errors we suggest to use the maximal lateral position error of 20 Nm (Nautic mile) given in [10]. In order to evaluate the stability condition above, we compute a two dimensional perturbation within this maximal

³ This again goes slightly outside of the scope of [9], which only considers deterministic algorithms. This experiment can however be useful to assess learning stability in a wide sense.

radius. Note that, for more complex use cases involving data in the form of text or image, the notion of “perturbation” is not so well-defined as in our use-case, and choosing the right notion of “perturbation” is already a challenge necessitating knowledge on the operational noise level and the Lipschitz constant of the targeted function (i.e., the local variation of f). A recent example of an expert definition of the maximal safe perturbation in subranges of the ODD can be found in [11].

Feasibility: On the IGRF use case, both the input space and the neural network have a small size. Therefore it is possible to employ complete formal methods (such as the SMT-based method described in [12]), to verify the stability property over all the input space⁴. For more complex problems, an estimation of ϵ can be empirically evaluated; either by sampling in the neighborhood⁵ of the test set samples, by using adversarial attack methods [13, 14] (aiming to maximize the error in the neighborhood), or by incomplete formal methods (such as abstract interpretation [15]). None of these methods provides guarantees for all x and x' , much less so when the ODD is high dimensional. For such more complex problems, the following probabilistic formulation of the property would be more convenient:

$$P_x(\forall x' \in B(X, \delta), |\hat{f}_S(X) - \hat{f}_S(x')| \leq \epsilon) \geq 1 - \alpha,$$

where X is a random point in the ODD (drawn from some distribution), and the ball $B(X, \delta)$ is the set of all $x' \in ODD$ such that $\|x' - X\| \leq \delta$. The above probabilistic property would mean that for most points x in the ODD (representing a fraction at least $1 - \alpha$ of the ODD), the surrogate model would not vary too much in a close neighborhood of x .

For the IGRF use case, we have experimented Monte-Carlo sampling estimation. We also perform an incomplete formal method (with Alpha-Crown [15]) on one thousand samples of the test set. This method is designed to compute upper u_p and lower l_p bounds containing model outputs (e.g. $l_p \leq \hat{f}_S(x') \leq u_p$), when the input is contained inside an l_p -ball around x : $B_p = \{x' \mid \|x' - x\|_p \leq \delta\}$ ⁶. We perform the computation with $p = 2$ for 10K samples and plot the absolute difference between these bounds and $\hat{f}_S(x)$.

Results are shown in Figure 4. Interestingly, on this use case, both methods (Monte-Carlo and Alpha-Crown) present coherent results and show that:

- the ML model is not stable close the north pole,
- the estimated model stability is consistent with the performance requirements.

⁴ This was not done in this study due to lack of time and resources.

⁵ A particular attention should be paid to the condition “inside the ODD”.

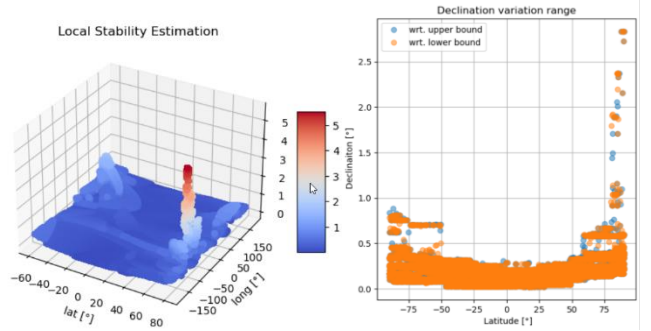


Figure 4 Trained model stability estimation: (left) Estimated local variations by Monte-Carlo sampling (right) Alpha-Crown upper and lower bound estimation for a 20Nm position perturbation [Note that only a subsample of test was processed with this method].

In conclusion, stability estimation is a pertinent tool to evaluate model vulnerabilities, when the variations of the ground truth function are known.

These results prove the feasibility of the objective in the low dimensional ML model that we study. However, the *clarity* and the *applicability* of the objective, such as the definition of the appropriate perturbation (e.g., δ) and model impact (e.g., loss function and ϵ), greatly depend on the use case and require both ML and operational expertise. Furthermore, since currently formal methods do not scale to large deep learning models, stability is mainly estimated by Monte Carlo or Adversarial methods, providing only a statistical lower bound estimation. Consequently, we cannot assume that this objective is *feasible* for such tasks.

E. Analysis of objective LM-13: Model robustness

The next objective is defined by:

Objective LM-13: The applicant should perform and document the verification of the robustness of the trained model in adverse conditions.

The concept paper suggests evaluating the model’s robustness against three types of examples:

- *Edge or corner cases* that can arise when considering data within the ODD but with one (resp. at least two) input variable(s) that is(are) close to the extremal values of the ODD;
- *Out of distribution (OoD)* examples that correspond to input data that are not covered by the training set distribution;
- *Adversarial* examples that may affect the AI/ML constituent expected behavior.

Edge, Corner and OoD. Concerning *clarity*, the definition of the edge, corner, and OoD examples is quite clear for our use case, but this definition is challenging for high-dimensional data.

As for *applicability and feasibility*: In the context of the surrogate use case, as we master data generation, it is quite easy to generate such test samples. For example, OoD samples will coincide with out of ODD samples (since the training dataset is drawn uniformly within the ODD). We can generate samples in the latitude range [82°N, 83°N] and evaluate the ML model

⁶ For the sake of simplicity, the study was done on an ℓ^2 -ball of the neural network input space. Since our neural network is in fact only provided with vectors of the form $(\cos(\theta), \sin(\theta), \cos(\phi), \sin(\phi))$ as inputs, this over-approximation leads to conservative stability estimates.

performances. Table 3 presents some results for 1000 samples, revealing, as expected, a performance degradation outside the ODD.

Latitude range	95% accuracy on OoD set
82°N - 83°N	11.60°
61°S - 60°S	8.81°

Table 3 OoD performances evaluation

We must highlight that for real-world use-cases, collecting corner, edge, and OoD points may be a challenge in itself. Detection of OoD samples is also a challenge for safety in order to monitor the usage of the ML model, but this is part of other objectives. *Feasibility* may not be reachable for some real-world cases, since OoD, edge and corner cases are not easily defined for high-dimensional data.

Adversarial robustness. Adversarial robustness is generally defined for classification tasks, with few works in the literature addressing regression. For regression tasks, a definition is provided in [16], which refers to the "worst perturbation" \hat{u} defined, for a given sample (x, y) , as:

$$\hat{u} = \operatorname{argmax}_{u: \|u\| \leq \delta} |\hat{f}(x + u) - y|$$

Estimating the worst perturbation \hat{u} seems *applicable and feasible* for our surrogate use case. However it must be noted that the "worst perturbation" for a given sample (x, y) will depend on the operational noise level and the Lipschitz constant of the targeted function (i.e., the local variation of f). Besides, the proposed definition prevents us from using many of the existing formal methods, such as [15].

To enhance *feasibility*, we propose to use the following tractable definition: Given a tolerated variation ϵ , find the largest perturbation norm δ (also called robustness radius) on x :

$$\max\{\delta \geq 0: \forall \|u\| \leq \delta, |\hat{f}(x + u) - \hat{f}(x)| \leq \epsilon\}.$$

With this definition, the previous results on trained model stability obtained with [15] or by Monte-Carlo sampling (Section IV.D) are applicable. This local largest perturbation should be compared to the knowledge of the target function to provide interesting features on ML model robustness.

F. Analysis of objectives LM-04 and LM-14: Generalisation bounds

Two LM objectives focus on the generalisation bounds of the trained model \hat{f}_S :

Objective LM-04: The applicant should provide quantifiable generalisation bounds.

Objective LM-14: The applicant should verify the anticipated generalisation bounds using the test data set.

Informally speaking, the generalisation ability of a trained model is about how well it performs on unseen operational data. This is formalized in the statistics and ML theory literatures (see, e.g., in [17] or [18]) through the statistical notion of *risk*.

The *risk* of a trained model \hat{f}_S , denoted by $R(\hat{f}_S)$, is the (theoretical) average error over all possible operational points, weighted by an appropriate distribution. In our case, since the ground truth is given by the output $f(x)$ of the IGRF-13 model, and since we consider uniformly distributed latitude θ and

longitude ϕ within the ranges 60°S-82°N and 180°W-180°E, the risk reads:

$$R(\hat{f}_S) = \int_{-60}^{82} \int_{-180}^{180} |\hat{f}_S(x_{\theta, \phi}) - f(x_{\theta, \phi})| \frac{d\theta}{142} \frac{d\phi}{360},$$

where $x_{\theta, \phi}$ denotes the Earth location at latitude θ and longitude ϕ (at an altitude of 100 meters). Importantly, the risk $R(\hat{f}_S)$ depends on the training set S ; it is a random variable.

A *generalisation bound* is a probabilistic bound on the risk $R(\hat{f}_S)$, typically expressed as a sum of an observed average error (called the empirical risk) and some statistical margin. Depending on whether the empirical risk is measured on the training set S or on some new calibration dataset S' , different mathematical tools are used. To the best of our knowledge, there are at least three families of methods to obtain generalisation guarantees.

A first family of bounds, which we could call *training-based generalisation bounds*, use the training set S to estimate the risk $R(\hat{f}_S)$ with the empirical risk given in our case by

$$R_S(\hat{f}_S) = \frac{1}{n_S} \sum_{x \in S} |\hat{f}_S(x) - f(x)|,$$

where n_S is the number of training examples. A rigorous statistical margin is then computed, i.e. a guaranteed upper bound on the *generalisation gap* $G = R(\hat{f}_S) - R_S(\hat{f}_S)$ that holds with high probability over the draw of the training set S . Various such bounds exist. They typically depend on the number n_S of training examples, on some (light) properties of the data distribution, and (to account for possible overfitting) on the model family complexity (e.g., the number of layers or parameters of the neural network, the type of activation function, etc). Unfortunately, such bounds are typically too large to be practical. A historical example in regression is given by the pseudo-dimension bounds (a generalization of VC-bounds to regression problems; see Theorem 11.8 in [18]). It is well known that these bounds are conservative (and thus typically pessimistic), as noted in the concept paper. Indeed these bounds control the generalisation gap $R(g) - R_S(g)$ of all models g under consideration (e.g., when varying all parameters of a given architecture) instead of the trained model \hat{f}_S only, and hold for virtually any data distribution.

Next we focus on *post-processing methods* that seem more promising in the near future. Such methods require a *calibration set* S' , which is a new dataset drawn independently from the training set S , and on which the trained model \hat{f}_S is either evaluated or modified (see below). Post-processing approaches typically yield better bounds than training-based methods, as they offer guarantees on the trained model only, instead of the whole model family.

Post-processing evaluation of \hat{f}_S . In this post-processing setting, the empirical risk is computed on the calibration set S' :

$$R_{S'}(\hat{f}_S) = \frac{1}{n_{S'}} \sum_{x \in S'} |\hat{f}_S(x) - f(x)|$$

where $n_{S'}$ is the size of S' . Then, the risk $R(\hat{f}_S)$ is upper bounded by $R_{S'}(\hat{f}_S)$ plus some guaranteed statistical margin. Various such generalisation bounds exist [17, 18] (using so-called concentration inequalities [19]). For example, when both outputs $\hat{f}_S(x)$ and $f(x)$ are bounded in $[0, 1]$, Hoeffding's inequality yields $P_{S'}\left(R(\hat{f}_S) \leq R_{S'}(\hat{f}_S) + \sqrt{\frac{\ln(1/\delta)}{2 n_{S'}}}\right) \geq 1 - \delta$, which means that the generalization bound

$$R(\hat{f}_S) \leq R_{S'}(\hat{f}_S) + \sqrt{\frac{\ln(1/\delta)}{2n_{S'}}$$

is valid for at least a fraction $1 - \delta$ of all possible calibration sets S' (while only one of them is observed in practice). Note that the bound is valid for any training set S .

Another example is given by Bernstein's inequality. The following version also holds when $\hat{f}_S(x), f(x) \in [0,1]$, for a fraction at least $1 - \delta$ of all possible calibration sets S' :

$$R(\hat{f}_S) \leq R_{S'}(\hat{f}_S) + \sqrt{\frac{2R_{S'}(\hat{f}_S)\ln\left(\frac{1}{\delta}\right)}{n_{S'}}} + \frac{2\ln\left(\frac{1}{\delta}\right)}{n_{S'}}$$

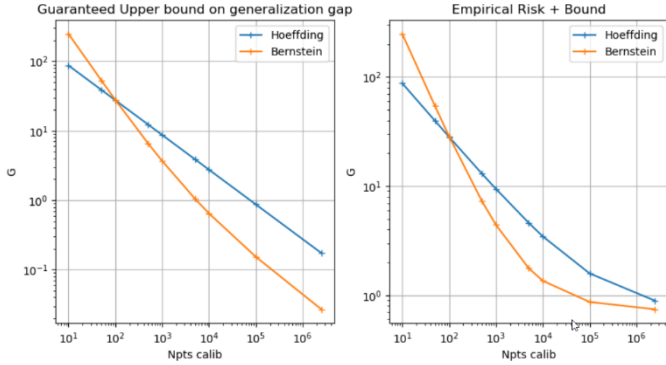


Figure 5: Hoeffding and Bernstein generalisation bounds in the IGRF use case.

In Figure 5 we plot the above two generalisation bounds in our IGRF use case (to that end, $\hat{f}_S(x)$ and $f(x)$ are first normalised to $[0,1]$, but the bounds are then converted back into degrees). On the left, we display the two guaranteed statistical margins

$\sqrt{\ln\left(\frac{1}{\delta}\right)/(2n_{S'})}$ and $\sqrt{2R_{S'}(\hat{f}_S)\ln\left(\frac{1}{\delta}\right)/n_{S'}} + 2\ln\left(\frac{1}{\delta}\right)/n_{S'}$ as functions of $n_{S'}$. On the right plot, we display the resulting generalisation bounds, given by the sum of the empirical risk $R_{S'}(\hat{f}_S)$ and these guaranteed statistical margins. They are statistically guaranteed upper bounds on the average magnetic declination error, for a latitude θ and longitude ϕ that are uniformly distributed within the ranges $60^\circ\text{S}-82^\circ\text{N}$ and $180^\circ\text{W}-180^\circ\text{E}$. A classical observation (from statistics theory) is that Bernstein's inequality entails a better bound, at least for sufficiently many calibration examples.

Post-processing modification of \hat{f}_S : risk-controlling prediction sets. An alternative family of post-processing methods consists in adding a predictive uncertainty quantification feature on top of the trained model. After modification, the predictor outputs a set of values (called a *prediction set*, typically an interval) instead of a single value, but with the guarantee of containing the ground truth with high probability. Conformal prediction methods [20] have gained renewed attention due to their simplicity and genericity. Next we focus on one algorithmic variant known as *risk controlling prediction sets (RCPS)* [21]. Just as before, this approach is only applicable once the model \hat{f}_S has been trained and requires an additional independent calibration dataset S' .

Several RCPS instances exist. For pedagogical purposes we describe a simple version below, which consists in replacing predictions $\hat{f}_S(x)$ with a prediction set $\mathcal{C}_\lambda(x) = [\hat{f}_S(x) - \lambda; \hat{f}_S(x) + \lambda]$. To that end, the user first defines a risk level α , and then computes a margin value $\hat{\lambda}$ by solving some optimization problem specified in [21]; roughly speaking, $\hat{\lambda}$ is

chosen so that the empirical risk on the calibration set S' plus some statistical margin (given by, e.g., Hoeffding's or Bernstein's inequalities) falls below α .

This process comes with a probabilistic guarantee, which in our surrogate use case reads:

$$P_{S'}(P_X[f(X) \in \mathcal{C}_{\hat{\lambda}}(X)] \geq 1 - \alpha) \geq 1 - \delta$$

This means that for a fraction $1 - \delta$ of all possible calibration sets S' , the prediction sets $\mathcal{C}_{\hat{\lambda}}(x) = [\hat{f}_S(x) - \hat{\lambda}; \hat{f}_S(x) + \hat{\lambda}]$ contain the ground truth $f(x)$ for most inputs x (at least a fraction $1 - \alpha$ of all inputs).

We apply RCPS to the IGRF use case, with $\alpha = \delta = 0.05$ and a calibration dataset of 10K samples for each latitude range. Results are shown in Table 4. The computed margin $\hat{\lambda}$ (3rd column) is consistent with the performance requirements (2nd column). Note that the 4th column somehow corresponds to the objective LM-14, but that our conclusions (3rd column) are slightly more conservative, as statistical wisdom suggests.

Latitude range	Performance requirements (95%) (°)	Lambda (°)	Observed 95% quantile (°)
-50°, 50°	2	1.7	1.51
50°, 73°	3	2.1	1.966
-60°, -50°	3	3.5	3.140
73°, 79°	5	3.9	3.472
79°, 82°	8	5.9	5.004

Table 4: Application of the RCPS method to the IGRF use case

In conclusion, back to our three criteria:







- *clarity*: we found that the LM-04 and LM-14 objectives are clear enough, though several interpretations are possible (cf., e.g., our two post-processing approaches).
- *applicability*: these objectives are applicable. Note that we had to assume some distribution on the latitude θ and longitude ϕ . For another distribution, the results in Table 4 would likely be different.
- *feasibility*: the objectives can be reached with post-processing methods, within reasonable computational costs, and with theoretical guarantees. The latter however crucially rely on the fact that the examples in the calibration set are independent and drawn from the right distribution (the uniform distribution in our case).

We stress that instances where the above guarantees are breached may be concentrated within specific segments of the ODD, which could significantly impact the integration of such metrics in safety assessments.

G. Overview of LM objectives analysis

The detailed analyses of the previous sections led to several conclusions regarding the clarity, applicability, and feasibility of the LM objectives under study. Table 5 below provides a synthetic overview of our results, with a focus on feasibility. The conclusions drawn pertain to the magnetic declination estimation use case. Though not showed in the paper, we also analyzed the LM objectives on other toy use cases, for regression (a univariate nonparametric regression problem with Gaussian noise) and for classification (the classical two moons dataset). We obtained similar conclusions, though these (non-surrogate) ML tasks raise additional challenges.

Table 5. Overview of the analyzed objectives.

Objective	DAL	Feasibility?
Performance on test set (LM-09)	D, C	 Empirically: Yes
Bias-Variance analysis (LM-07, LM-08)	C	 Empirically: only rough estimation of bias and variance. Computationally prohibitive. Bias estimation is mostly specific to the surrogate setting (known $f(x)$). Formally: No
Learning algorithm stability (LM-11)	C	 Empirically: Only rough approximation Formally: No, very limited existing Theory
Trained model stability (LM-12)	D, C	 Empirically: Yes with average metrics Formally: Yes, formal methods for specific model architectures (not scalable to higher dimensional problems).
Model robustness (LM-13)	D, C	 Empirically: Yes but clearer definitions and metrics needed
Generalization bounds (LM-04, LM-14)	C	 Formally: No for training-based bounds: computable but not actionable. Yes for post-processing bounds. Warning: these bounds require statistical properties on the datasets.

V. DISCUSSION

In Section IV, we only address the technical challenges raised by the LM objectives, namely: the clarity of the objectives in terms of their mathematical definitions, the applicability to the IGRF use case, and the feasibility (computational cost, choice of some parameter values, theoretical guarantees, assumptions on the data, etc). However, we do not address the link between system safety and the LM objectives. This connection is established within the concept paper for LM-04 and LM-09 (as recalled in Section II), but in the future it would be useful to re-assess and refine this link. The contributions of the other LM objectives to system safety also need to be thoroughly investigated.

In Section V.A, we recall the paradigm shift from programming to learning, since it has key consequences on safety assurances. In Section V.B, we raise several questions concerning the contribution of safety assurance to system safety.

A. The paradigm shift

The concept paper proposes to address "the paradigm shift from programming to learning" with "learning assurances". We remind that the goal of the assurances is to obtain as many guarantees as possible that the contribution to safety of residual errors during operation will be acceptable. We highlight below how the two main engineering approaches described here (human programming-based and machine learning-based) address this goal radically differently. We conclude that they are significantly different information processing (i.e., transformation) approaches. In this perspective, we depict here the following fundamental differences of this "shift":

1. **Actor of the transformation.** To minimize the errors made by humans, engineers rely on well-established principles and practices, supported by strong evidence gathered throughout extensive experience. On the other hand, to minimize the errors done by the machine, the applicant can only rely on a deep understanding of the learning process.

⁷ Even in the case of surrogate models, for which detailed specifications of the function may be available, the compression task performed by the model cannot be fully specified. Note that

2. **Complexity of the problems to be solved.** We make the rather obvious assumption that ML techniques are to be used whenever no efficient alternative solution exists (i.e., one which can be completely specified and coded by humans)⁷. Consequently, no individual (or group of individuals) can analyze and verify exhaustively whether the computations of the ML-based software are correct or not. In most cases (in particular, when formal methods do not apply), engineers can only perform an empirical analysis of the ML model on some finite set of test examples, as if it were a black box. The human-written code based on complete software specifications can, on the other hand, be fully verified by other humans.
3. **Intrinsic nature of the transformation process.** In the case of human programming, software requirements are transformed into code via a succession of abstractions and decompositions, from the highest and widest level, to the lowest and thinnest one. This transformation allows several intermediate verifications, by either tests or analysis, and is end-to-end understandable and traceable. In the case of ML software, the transformation (i.e. the learning phase) is mostly done by an optimization algorithm, which computes the parameters of the ML model, by minimizing a loss function to automatically capture statistical patterns in the training data. These are two fundamentally different ways of processing the information.
4. **Coverage of the input data.** ML is mostly used to solve highly dimensional problems, which are impossible to describe / specify completely. Therefore, ensuring an exhaustive coverage of the input data space through massive testing is prohibitive for ML software (in absence of strong hypotheses regarding the data or the model). On the other hand, extensive coverage tests of human-written software are far more feasible essentially with the help of "equivalence classes" methods. The concept of "equivalence classes" frequently used in classical software test practices does not apply to ML software, due to the incomplete nature of the specifications of the problem being solved.

B. Safety-related challenges

We now briefly discuss important safety-related challenges that arise from the aforementioned paradigm shift. In safety, the main goal is to identify foreseeable failures and to obtain as many guarantees as possible that the impact and likelihood of failures will be acceptable in operation. It is thus important to question the link between the satisfaction of the LM objectives and this safety principle. Though these objectives appear to be very intuitive at first sight, we anticipate that seemingly small technical details in their instantiation might influence safety conclusions in a non-negligible way. Let us mention several examples, which appear at different levels.

1. **When interpreting an LM objective in terms of a mathematical definition.** For example, for the IGRF use case, in Section IV.F we provide two generalisation guarantees, but only one of them seems directly related to safety or, more precisely, to the performance requirements

is a reason to forbid compression options in the compilers in the safety critical software.

given in [2]⁸. Indeed a small risk $R(\hat{f}_S)$ only means that the average absolute error over the Earth's surface (for a specific distribution) is small, which does not directly translate into whether the 95% performance requirements of Table 1 are met⁹. On the other hand, the RCPS method yields results that can be directly compared to these requirements; see Table 4.

2. **When applying a mathematical definition that depends on parameters, metrics, assumptions, etc.** Since the link between the LM objectives satisfaction and system safety is not clarified, the applicant can have trouble in motivating the choice of some parameter values (such as the β , δ or ϵ parameters in Sections IV.C and IV.D), performance metrics (the loss function involved in the risk definition), acceptable performance values, and data assumptions.

For the IGRF use case the absolute error seems to be the most natural metric choice, but this is use-case specific. The choice of parameter values, and how they contribute to system safety, also seems very delicate. For example, as discussed in Section IV.D, a very stable trained model might feature a poor accuracy. Therefore, while a too small δ for a given value of ϵ could be detrimental to safety (since the ML model could be sensitive to adversarial attacks), a too large δ may lead to inaccurate predictions and could be detrimental to safety too.

Note from the previous paragraph that maximizing both robustness and accuracy is virtually impossible. This phenomenon contrasts with traditional assurance rules on software development. Indeed, traditional assurance rules can be cumulated to reduce the residual risk *i.e.*, the effects of a given rule will not cancel out the effects of another rule. The experiments conducted in this paper reveal that, when interpreted with our mathematical definitions with some parameter values, some LM objectives could be competitive. The classical cumulative property no longer holds for this specific phase of the ML development process. In other words, the Rearson metaphor of Swiss cheese slices does not apply anymore. In practice, whether all objectives can be satisfied simultaneously or not will depend on parameter values as well as other choices (e.g., performance metrics), which should thus be properly linked to system safety.

Overall, important efforts are needed to establish the links between the LM objectives satisfaction and system safety. This will enable to refine such objectives (in terms of mathematical definitions, parameter values, performance metrics, acceptable performance values, data assumptions, etc), or possibly to define new LM objectives.

VI. CONCLUSIONS

In this paper, we analyze several of the objectives proposed in [1]. We would first like to acknowledge all the structuring efforts towards the challenging goal of certifying safety-critical systems with AI components. However, our study shows that, even on a seemingly simple use-case, these objectives raise a

⁸ As noted in Section II, though the IGRF use case may not strictly fall under the EASA guidelines for critical airborne systems, it presents a realistic, well-defined, and thoroughly documented system. We use it as an illustrative example here.

⁹ If the risk $R(\hat{f}_S)$ were redefined for each latitude range of Table 1 (instead of a global average), Markov's inequality

series of technical and methodological challenges; see Section IV.G for a synthetic overview. While intuitive and arguably helpful to gain confidence in ML-based systems, some of these objectives turn out to be ambiguous or unfeasible from a practical standpoint in the analyzed context. Satisfying these objectives for non-surrogate ML tasks, or quantifying their eventual (degree of) satisfaction to the reduction of safety-related risks may posit additional hard challenges.

In light of these findings, we consider that:

- **Further academic research** must be conducted to develop methods that guarantee trustworthiness of an ML constituent. The scientific literature contains few appropriate methods that allow for the straightforward and efficient verification of the above objectives.
- Despite the relevance of the guidelines towards the certification goal, the scope and formulation of **several requirements should be refined and clarified**. This clarification is key to address complex use-cases.

Acknowledgements. This work benefited from the AI Interdisciplinary Institute ANITI, which was funded by the French "Investing for the Future--PIA3" program under the Grant agreement ANR-19-P3IA-0004. The authors gratefully acknowledge the support of the DEEL project.¹⁰

VII. REFERENCES

- [1] EASA, "EASA concept paper: first usable guidance for level 1&2 machine learning applications," 2024.
- [2] EASA, "Easy Access Rules for Large Aeroplanes (CS-25)," revision January 2023.
- [3] P. Alken, E. Thébault, C. D. Beggan and al, "International Geomagnetic Reference Field: the thirteenth generation," *Earth Planets Space*, vol. 73, no. 49, 2021.
- [4] "pyIGRF: IGRF-13 Model by Python," [Online]. Available: <https://pypi.org/project/pyIGRF/>.
- [5] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *Annals of Statistics*, vol. 7, pp. p. 1-26, 1979.
- [6] EASA and D. AG, "Concepts of Design Assurance for Neural Networks (CoDANN)," 2020.
- [7] P. Domingos, "A unified bias-variance decomposition and its applications," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, Stanford, CA, USA, 2000.
- [8] G. Valentini and T. G. Dietterich, "Bias-Variance Analysis of Support Vector Machines for the Development of SVM-Based Ensemble Methods," *Journal of Machine Learning Research*, vol. 5, pp. 725-775, 2004.
- [9] O. Bousquet and A. Elisseeff, "Algorithmic Stability and Generalization Performance," in *Advances in Neural Information Processing Systems*, 2000.
- [10] ICAO, Performance Based Navigation, 3rd edition, 2008.

would imply high probability results similar in spirit to Table 4, but this implication would be crude. The RCPS method addresses probabilistic bounds directly, in a mathematically tighter way.

¹⁰ <https://www.deel.ai/>

- [11] M. Ducoffe, G. Povéda, A. Galametz, R. Boumazouza, ., M.-C. Martin, J. Baris, D. Daverschot and E. O'Higgins, Surrogate Neural Networks Local Stability for Aircraft Predictive Maintenance, 2024.
- [12] G. Katz, C. W. Barrett, D. L. Dill, K. Julian and M. J. Kochenderfer, "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks," *CAV*, 2017.
- [13] J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.
- [14] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *ieee symposium on security and privacy (sp)*, 2017.
- [15] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin and C.-J. Hsieh, "Fast and Complete: Enabling Complete Neural Network Verification with Rapid and Massively Parallel Incomplete Verifiers," in *ICLR*, 2021.
- [16] K. Gupta, B. Pesquet-Popescu, F. Kaakai, J.-C. Pesquet and F. D. Malliaros, "An Adversarial Attacker for Neural Networks in Regression Problems," *IJCAI Workshop on Artificial Intelligence Safety (AI Safety)*, 2021.
- [17] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.
- [18] M. Mohri, A. Rostamizadeh and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed., MIT Press, 2018.
- [19] S. Boucheron, G. Lugosi and P. Massart, *Concentration inequalities: a nonasymptotic theory of independence*, Oxford University Press, 2013.
- [20] V. Vovk, A. Gammerman and G. Shafer, *Algorithmic Learning in a Random World*, 2nd ed., Springer-Verlag, 2022.
- [21] S. Bates, A. Angelopoulos, L. Lei, J. Malik and M. Jordan, "Distribution-Free, Risk-Controlling Prediction Sets," *Journal of the ACM*, vol. 68, no. 6, 2021.
- [22] M. Ducoffe, S. Gerchinovitz and J. Sen Gupta, "A high-probability safety guarantee for shifted neural network surrogates," in *SafeAI 2020*, 2020.