Perspectives on AI-ML Safety Assurance

Emmanuel Ledinot, Philippe Quere, Philippe Baufreton, Jean Gassino, Franck Serratrice, Hugues Bonnin, Damien Chabrol, Amina Mekki-Mokhtar, Olivier Appere, Joseph Machrouh

Perspectives on AI-ML Safety Assurance

Emmanuel Ledinot
emmanuel.ledinot@thalesgroup.com

Jean Gassino
jean.gassino@irsn.fr

Amina Mekki-Mokhtar
amina.mekkimokhtar@ansys.com

Philippe Quere
philippe.quere@stellantis.com

Franck Serratrice
franck.serratrice@renault.com

Olivier Appere
appere@adacore.com

Philippe Baufreton
philippe.baufreton@safran.com

Hugues Bonnin
hugues.bonnin@continental.com

Joseph Machrouh
joseph.machrouh@thalesgroup.com

Damien Chabrol
damien.chabrol@kronosafe.com

***Abstract*— AI-ML suffers from a reliability glass-ceiling phenomenon (e.g. ~10^{-3} error/inference), making it incompatible with safety-criticality. Several orders of magnitude are missing. We explain why, we point to the characteristics of ML that conflict with the assurance objectives assigned to safety-critical developments. Could encapsulation of ML constituents into fault-tolerant architectures, ML development assurance, and software/hardware development assurance, altogether mitigate the gap? We argue that in spite of impressive progress of ML state-of-the-art, the answer is negative. Drawing from Topological Data Analysis (TDA) and set-based non-linear control, we propose to supplement ML *point*-based specification and verification with *volume*-based specification and verification to meet 10^{-5} err./ inf. levels, as a minimum. We outline the rationale of a new research field we name (Ultra) Reliable Machine Learning, at the confluence of TDA, statistics on manifolds, and ML safety assurance. Some cross-domain safety regulation principles guide the underlying rationale. We illustrate the methodology on image classification.**

Keywords— Machine Learning, ML reliability, Safety assurance, ML assurance, latent manifold, Topological Data Analysis, persistence homology, extensional coverage analysis.

I. INTRODUCTION

Data analysis and statistics have first developed to extract synthetic information from population data as *insights* on complex phenomena (descriptive statistics). Inferential statistics then focused on explanatory models of past observations, to get predictors on some limited aspects of complex phenomena. Never until recently, had statistical estimation to address safety-critical 'control'. We use 'control' in the broad sense of OODA loops (Observation, Orientation, Decision, Action), where control of physics is involved and life, goods or environment is at risk.

Machine Learning, especially Deep Learning (DL), opened a new era: unprecedented performance in machine vision and problem solving in high dimension. However, chaotic behavior exemplified by adversarial examples limited DL applicability [39], and is still a matter of concern. Could DL-based components, developed with extreme rigor and encapsulated in fault-tolerant architectures, deliver services that meet the reliability requirements specific to safety-critical 'control'? This type of requirements is new to Machine Learning and data science.

The co-authors of this paper are members of the Embedded France association's working group dedicated to analysis of safety assurance standards in safety-related industrial domains, to contribute their evolution [24]. We investigate the case of Machine Learning in this paper since ML-dependent safety-criticality is now on the agenda of aeronautics [2] and of automotive industry. Our focus is limited to ML *reliability*, ML *verification*, and to safety assurance of ML-dependent systems.

To our knowledge, current best accuracy scores on the easiest of image classification benchmarks (MNIST) are about 2.10^{-3} error/inference [40]. From a system safety perspective, this reliability level is poor: one error every seven lines containing 80 digits each. To make the gap more explicit, let us assume a 50Hz input stream of digits processed by an AI-ML-dependent safety-critical vision-based controller. It would make ~360 generalization errors per hour, when reliability target in the most critical case discussed in this paper would be one every billion of hours.

To address this gap, [1] screened the techniques amenable to improve ML reliability. They questioned the feasibility of reaching the reliability levels required by highest DAL[1]s and concluded negatively. After some scoping and terminological preliminaries, we summarize this survey of reliability augmentation methods. We propose a *conjectural* explanation why the reliability enhancement attempts uniformly failed (sections II, III, IV).

Then, we discuss why software assurance will have no impact on this reliability gap (section V), and why fault-tolerant architectures will solve only the easy cases (section VI). At this stage, we conclude that for true ML-dependent safety-criticality, there is no escape from improving ML reliability by several *orders of magnitude*.

From a geometric and topological perspective on approximant adjustment, we convey intuition on how great the challenge is. Thanks to recent advances in Topological Data Analysis (TDA), we propose a research path that would control ODD [2] modeling, data sampling, generalization domain definition, and approximant adjustment more tightly than achieved today. We review some recent papers that suggest relevance of such an attempt. We compare the rationale of safety-critical software verification, with our TDA-enabled (U)R-ML verification proposal (section VIII).

[1] Development Assurance Level

[2] Operational Design Domain, see Road vehicles — Safety of the Intended Functionality ISO 21448 standard.

Finally, we discuss whether ML-dependent safety-critical 'control' could reach the ultimate reliability level of 1, i.e. *correctness*. Software engineering and assurance managed to ensure extremely high levels of quality. We compare the two domains on specification and verification.

Contribution: We propose a diagnosis on the ML-reliability plateau. We propose orientations to overcome the reliability gap by supplementing current point-based approach of data science with a TDA-enabled *volume*-based approach.

Disclaimer: The views expressed in this paper are those of the authors as members of the Embedded France Working Group on safety assurance standards. They may not reflect the opinion of their affiliations.

II. SCOPING AI-ML-DEPENDENT SAFETY

A. Systems perimeter

We address ML-dependent safety-critical systems. Since our group is cross-domain, for the rest of the paper we use the following convention: DAL A is an abbreviation of all the corresponding assurance levels in the other industrial domains. DAL A stands for DAL A (aeronautic), ASIL D (automotive), SIL 4 (railway, process industry and many domains) and class 1 (nuclear).

In this paper, an ML-component is classified as safety-critical if, and only if, it is a "Single Point of Catastrophic Failure" (SPCF). In other words, some error, in adverse foreseeable conditions, could lead to a catastrophic accident. DAL A is mandatory for SPCF components: no mitigation mechanism in the system architecture to prevent some failure causality chain originating from the ML-component to evolve into a catastrophic accidental scenario. We abbreviate "SPCF-ML" such situations.

Our prototypical SPCF-ML example in automotive is pedestrian detection systems coupled to automatic-braking systems. See [29] for state of the art on DL-dependent pedestrian detection performance: robustness and accuracy are still a major concern. In aeronautics, inhabited autonomous urban air mobility is the example we have in mind. More generally, we consider ML-dependent vehicle control, safety-critical healthcare devices, and all kinds of safety-critical operational technologies (OTs).

B. ML perimeter

We consider off-line supervised learning in high to very high input-space dimension (e.g. 10^4 to 10^6 and beyond). We exclude continuous learning and recent ML developments like transformers and LLMs. Regarding the ML-safety survey [5], we address Robustness and Monitoring. Ethics and Alignment are out of the scope of this paper.

C. Machine-vision perimeter

Open world semantic scene segmentation is the natural long-term goal. However, we do not claim supplementing such complex ML developments with TDA at first. In this paper, we limit ourselves to development and assurance rationale of a proof of concept based on MNIST[3]. 10^{-5} err./inf. is our first milestone to fill the reliability gap. We present it as an illustrative example of a generic methodology expected to be progressively scaled up to ML processing pipe-lines as complex as 3D scene segmentation. After MNIST [35], the planned next step is LARD (Landing Approach Runwaw Detection) [25]. Only then, could one conclude on (U)R-ML practical viability. MNIST and LARD have in common existence of strong knowledge on the data generation process that enable structured data interpretation.

III. TERMINOLOGICAL PRELIMINARIES

We need to avoid misinterpretation on terms like 'dimension', 'dimension reduction', 'latent' and a few more.

A. Machine learning

- *Approximant***,** any function $\mathbb{R}^n \rightarrow \mathbb{R}^p$, estimator of an underlying function specified by textual requirements and labeled datasets. We use 'ML-model', after adjustment, as synonymous of fitted approximant.
- *Inference***,** and generalization, are used as synonymous: approximant activation on some input vector not seen during the training, calibration, and testing phases.
- *Ambient space***,** also named embedding space: space where the vectors (or points) of the datasets spread. Depending on the context, we use "ambient space" for input only (nD)[4], output only (pD), or input-output $((n+p)D)$ space. For greyscale image classifiers, n is the number of pixels and p that of classes (e.g. MNIST: n=28x28=784, p=10).
- *Latent space* or latent manifold, the regions of the ambient space where the dataset points concentrate, i.e. cluster. Latent space has its own dimension named *latent* dimension, or *intrinsic* dimension.
- *Dimension reduction***.** The classical interpretation of this term is identification of the input space features that prominently condition the form of the output latent manifold (projection on a lower dimensional space keeping most of the information, like PCA[5]). We never use this meaning. We consider ambient to latent dimensionality collapse by shifting from an external view to an internal view of the point cloud. When continuous natural processes generate data, dimensionality collapse occurs. Physical, operational, and control laws constrain input, state and output data to concentrate in low-dimensional regions that unfold, split, curl, merge etc. in ambient space. (Manifold Hypothesis (MH) on point clouds [11]).

B. Logics

- *Extensional* refers to extension as defined in "Extension Theory" [6], i.e. vector encoding of magnitudes for geometric and algebraic calculation. In the sequel, we regard geometric and topological analysis of point clouds in vector spaces as synonymous with "extensional approach".
- *Intensional* qualifies definitions of sets or objects by *symbol* sequences (logical formulas, analytical expressions, characteristic predicates etc.). For

[3] MNIST is a prominent entry point benchmark in image classification community. It consists of 70000 handwritten digits elaborated by NIST in the USA.

[4] nD stands for n Dimensions (1D curves, 2D surfaces, etc.)

[5] Principal Component Analysis

example, first-principle models are intensional characterizations of process behaviors. Structural coverage in software testing is intensional. It is hooked to programs' source or binary code symbols. Ontologies of ODDs and analytic formulation of data-augmentation processes are on the intensional side as well.

IV. ML-RELIABILITY GLASS CEILING

A. *Reliability augmentation techniques*

In [1], a group of researchers investigated the means to improve ML reliability. Though ML made major progress on accuracy over the last two decades (1 to 2 orders of magnitude), 10^{-3}/inf. is still too poor from a safety engineering viewpoint. [1] reviews quantitative reliability results obtained by model diversification, by monitoring (ODD, robustness, I/O consistency), by robustness enhancement techniques (model stability and training stability), by selective classification, by conformal prediction, and by temporal redundancy on sequences.

Their main conclusion is the following: all the methods that tried to increase reliability by redundancy of independent models, i.e., models resorting to independent approximant spaces, independent datasets and independent optimization processes, succeeded only *marginally*. Reliability stayed stuck in the range of 10^{-2} / inference instead of the expected $10^{-4} = 10^{-2} * 10^{-2}$ or even $10^{-6} = 10^{-2} * 10^{-2} * 10^{-2}$. Moreover, these techniques improved reliability at the expense of significant availability losses.

B. *Common Cause Analysis*

Strong correlation of inference errors between independently developed ML-models, i.e. lack of independence between redundancies, is an experimental fact evidenced by [1]. It is consistent with [39] where evidence is given that an adversarial example designed for model1 trained and tested on dataset1 still fools model2 specifically developed to be *independent* of model1 (datasets, approximant space, and optimization process). Similarly, [38] demonstrated a limited 13% reliability progress. It is negligible from a safety engineering perspective given the reliability targets mentioned previously.

Since in this paper we are going to compare ML and software engineering in the safety-critical case, we recall that in the 1980s [37] evidenced experimental reject of the independence hypothesis on N-version programming.

What could be an explanation? Our working hypothesis that motivates our interest for TDA-augmented ML is that complexity of the latent manifold's shape could be the common mode that correlates error occurrences between the so-called "independent" redundancies[6].

Fig. 1. Model adjustment to a point cloud (green shape adjusted to the red spots). The dashed ellipses delineate topologically complex regions that are hard to fit correctly.

State space complexity of non-linear dynamical systems (attractors, curvature, holes, cavities, etc.), compelled control engineers to start by splitting it into covering subspaces where dynamics regime has some homogeneity and regularity amenable to a local linear approach. Then, they aggregate these local controllers into a unique global controller by mode switching and scheduling logics, up to complete coverage of the topologically complex reachable input/state/output space. The ML components we consider in this paper address the same type of continuous data manifolds. By contrast, standard data science addresses training datasets all at once, straight away at global scale.

Possibly, the ML model redundancies used in [1] failed to adjust reliably on the same topologically complex regions. Hard-to-fit regions of input space are *problem* dependent. In other words, they are ML-model *independent*, so they can correlate any pair of redundancies. The shape of training datasets is a potential *common cause* in ensemble learning.

C. *Plateauing performance*

When the approximant space is defined by the solutions to (n – 1) polynomial equations over n variables, the ambient space is nD and the latent space is 1D algebraic curves. Given k points in nD Euclidian space, finding a polynomial curve that links the k points is still an open mathematical problem [9]. By 2022, a proof of existence was published on the Web. It is under peer-review. In case of confirmation, more than a century will have been necessary to solve the (n-ambient, 1-latent) case for an intensively investigated class of functions.

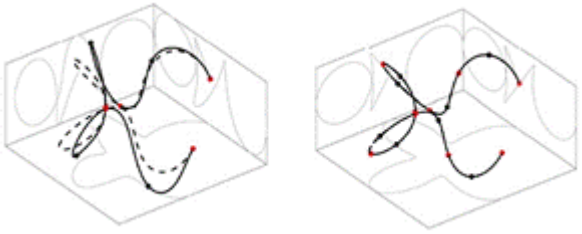

Fig. 2. The picture is courtesy of [8]. 1D latent manifolds in 3D ambient space. Limiting generalization errors to very small number of occurrences requires controlling adjustment with extreme precision. Impact of "fitting" variability on the 3 projected curves when "adjustment" varies slightly (difference between the dashed and non-dashed curves).

Admittedly, equation solving (i.e. 'exact adjustment') is of different nature than ML-model fitting. It is harder because of equation solving exactness. However, precision-controlled fitting in high dimension is a very difficult problem as well, even if a "flexible" one[7]. We advocate that high reliability of generalization will necessitate sophisticated mathematical tools to control where and *why* generalization errors occur. The ability to explain why a generalization error occurred in order to fix it will be mandatory for DAL A ML. Any known error that could potentially be a single cause of catastrophic failure, should be eliminated to comply with regulation.

D. *Zero-measure verification*

Any behavioral specification defined by a cloud of points is extremely poor with respect to:

- The immensity of the high dimensional ambient space,
- The shape complexity of the input and input-output latent manifolds.

[6] In section X, another potential cause is considered on MNIST: labeling errors [41].

[7] Because the inverse problem is ill-posed.

Meeting inference failure rates as low as 10^{-k} err./inf. , $k \geq 5$, is highly demanding. Sample-oriented by nature, statistical functional estimation naturally relies on point-based verification. Extensional verification coverage by the end of cross-validation, i.e. the covered volume of behavior, by means of some N-point testing dataset is N*0=0. In other words, the coverage is null because each point has no extension. At the opposite, the nD volume of the latent input manifold over which the estimated function should generalize reliably is gigantic and nearly devoid of specification information. We illustrate the specification miss on the MNIST classification problem, and how TDA could help (section X). Worse, the generalization domain over which one should estimate probabilities of misclassification events is *undefined*. No integration, i.e. error counting, without specified integration domain, i.e. *defined* inference domain, and without *error-oracle* covering it exhaustively. Such error-oracle is named actionable specification in [30].

From safety engineering and assurance points of view, there is a discrepancy between on one hand the absence of explicit input-domain definition, the gigantic space where specification misses, the limited control of adjustment, and on the other hand the extremely demanding reliability levels required to get certification approval on ML-dependent safety-critical systems.

V. FILLING THE GAP WITH SOFTWARE ASSURANCE

Could the reliability plateauing problem (~10^{-3} err./inf.) be mitigated by implementation of ML-models with extreme rigor, i.e. with DAL A assurance level? The reason would be, following some misconceptions about development assurance, that DAL A developments deliver high integrity software, and accordingly that high integrity software would ensure 10^{-k} failure/h reliability levels, for values of k ranging from 5 to 9, depending on industrial domains.

The goal of software assurance is to ensure fidelity of the transformation process that converts system functional specifications like ML-models (e.g. TensorFlow mathematical equations) into binary code instructions. Fidelity, also named implementation *correctness*, or *compliance* or *semantic invariance*, means ensuring *extensional* behavioral equivalence between some ML-model and its executable object code counterpart. On the intensional side, the transformation of symbol sequences is complex. Preservation of the defined behavior is at risk. Regarding reliability of inference, DAL A ensures high trust on *reliability invariance* from model to executable object code, i.e. "garbage in, garbage out". It does not ensure reliability *augmentation* (e.g. up to 10^{-9} err./inf.) during the transformation process.

Explaining why there is no reliability augmentation provided by assured software is *not* discrediting the value of software assurance. Software assurance prevents introduction of flaws in the behavior-preserving symbolic transformation. One may found more information on the link between qualitative and quantitative aspects of development assurance in [10]. In particular, the domain-dependent relationships between reliability levels k and assurance levels (A, B, C, D) are *conventions* that associate qualitative leveling of rigor with expected reliability in case of residual faults. Assurance splits trustworthiness construction in two policy regimes (cf. section VII). It needs some correspondence between the two for global consistency. This correspondence is not a *convertibility rule* between fault-prevention rigor levels, i.e. DALs, and reliability levels. However, return of experience over ~50 years demonstrated validity of these conventions.

VI. FILLING THE GAP WITH SYSTEM FAULT-TOLERANT ARCHITECTURES

We consider the case of catastrophic failure dependent on the performance premium *uniquely* delivered by Deep Learning. For pedestrian collision avoidance systems or autonomous air taxis, Deep Learning has by far outperformed the classical and certifiable algorithms of computer vision. If some classical underperforming algorithm is sufficient as safety monitor to keep controllability in fault detection-isolation-recovery phases, then the DL-dependent channel provides only performance bonus. Form safety architecture point of view there is no true criticality assigned to AI-ML.

We extensively discussed in the group whether software engineering and assurance managed over time to prove sufficient effectiveness so that SPCF software was introduced in safety-critical architectures. Answer was yes, for aeronautics, space, automotive and railway. Nuclear is the exception (DAL B at most). We have no representative of medical device industry in the group.

In aeronautics for instance, in flight control systems in particular, there are architectures, functions, and limited regions of the flight domain where a specification flaw or an implementation error may constitute a single point of catastrophic failure. DL-dependent vision-based control for air taxis or pedestrian collision avoidance will lead to true SPCF-ML constituents as surely as it was the case for software. In the "no-backup" situations that define DAL A, extreme reliability is required and even perfect reliability named *correctness.* This is the motivation of our research program proposal on TDA-enabled (U)R-ML.

VII. ELEMENTS OF ASSURANCE PRINCIPLES

We review the foundational aspects of development assurance that interact with ML characteristics in the safety-critical case. We start with the rationale that splits assurance in two policies: correctness and rareness. In the sequel, we use 'quantitative' objectives exclusively for probabilistic quantification of event occurrences. As an example, 100% DC coverage, though 100% is a quantity, is not a quantitative assurance objective, in our sense at least. It is a software testing termination criterion dependent on a numerical value that conditions intensional cover.

A. Correctness .vs. rareness policies

Historical perspective helps understanding the split between fault prevention/elimination on one side, and probabilistic quantification of feared failure events on the other side. The former is applied to software and hardware development. The latter is applied to physical failure modes and their cascading effects. We quote the following text from aeronautical regulation to prove that probabilistic quantitative arguments were not primal in trustworthiness demonstrations. Logical, argument-based demonstrations of safety, even when software was absent (i.e. electromechanical systems), preceded probability-based evidences.

Design and implementation correctness of fail-safe mechanisms in charge of passivating the single points of catastrophic failures was the first and primary safety assurance

objective in aeronautics. It was the origin of the fault prevention process-based assurance methods.

> CS 25.1309 « Equipment, Systems, and Installations » AC251309-1B
> *"**In the early years** of aviation, airplane systems were evaluated to specific requirements: to the "**single fault**" **criterion**, or to the fail-safe design concept, which are explained below. As **later**-generation airplanes developed, their designers added more safety-critical functions, which generally resulted in an increase in the complexity of the systems designed to perform these functions. A safety-critical function was a function whose failure would result in a catastrophic accident. The potential hazards to the airplane and its occupants, in the event of failure of one or more functions provided by a system, had to be considered, as did the interaction between systems performing different functions. To assess the safety of a complex system—and the adequacy of system redundancy to meet the fail-safe criterion—the **FAA began assigning** statistical **probabilities to system failures** in AC 25.1309-1, dated September 7, 1982."*

Probabilistic assurance goals were introduced for the reasons explained in the verbatim, but the first accepted means of compliance were qualitative. Arguments of assurance cases were similar to that of qualitative physics applied to conservative approximations of failure propagation through system architectures. Orders of magnitudes were enough, and (causal) independence hypothesis between component and function redundancies were the primary concerns. Then, came computer-intensive probabilistic calculations and their acceptance as means of compliance (e.g. fault-tree analysis and Markov chain models).

Over a few decades, some unconscious cognitive bias spread in the safety engineering community. It consisted in reducing safety assurance goals to probabilistic ones, and probabilistic arguments to quantitative ones.

As software or hardware items, ML implemented models are deterministic artefacts. Nonetheless, as result of an engineering process they are realization of a random variable, valued by a mathematical function. The seeds of randomness are data sampling and stochastic features in adjustment algorithms. By extension, one could add as seeds of randomness, the model instability sources related to ill-posedness of the inverse problem, and addressed by the stability assurance objectives.

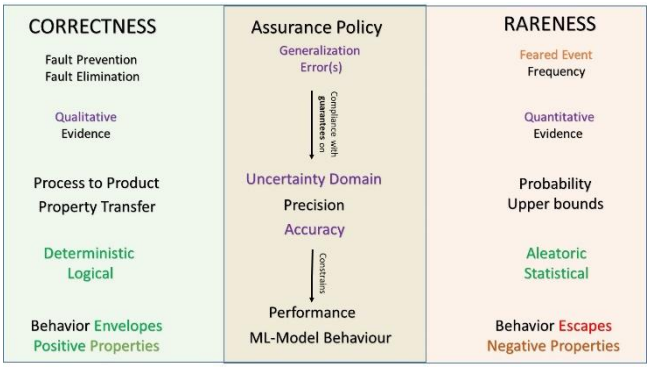

Fig. 3. Contrasting the two assurance policies. ML is amenable of both (overlay of green and amber). Preliminary to fig. 4 on status of SPCF generalisation errors.

Quantified sufficient rareness of ML-component failure modes would be the natural choice as assurance objective. We discuss this option in F. We tried to map the contrastive characteristics of the two assurance policies and their intricate relations with ML assurance in the following figure.

B. Actionable specifications

We support the analysis in [30] that singles out point-based specification as the prominent difficulty for ML safety engineering. We reuse the term "actionable". We interpret it as "amenable to computational evaluation" and consider it as equivalent to the 'perfect oracle' notion of [32]. Software (resp. hardware) testing of the implemented ML model, formal verification, and probabilistic quantification of error events, all need a computable oracle to decide whether model's response on input vector deviates from the intended, as specified.

The specified may diverge from the intended if needs capture is not correct and complete. The specified is pivotal for the following assurance objectives:

1. *derivation* of implementation from specification,
2. *correctness* of implementation w.r.t. specification,
3. *quantification* of failure modes.

Computer-decidability (test oracles, failure-mode oracles) of the specified is necessary for both assurance regimes. There are ambiguity cases in image classification where even human-decidability is not ensured. Another source of oracle miss is lack of ground-truth, quite common in ML application to perception systems. Safety engineering and assurance are severely hampered by miss of deviation oracles. ML assurance should exclude SPCF-ML in such development conditions.

C. Implementation derived from specification

Mitigation of complexity-induced risks by decomposition of the specified, by piecewise refinement, and by progressive and traceable derivation of implementation constructs from specification traits, constitutes a cornerstone of assurance. It is a "divide & conquer" error-prevention strategy to cope with error-friendly complexity.

A second cornerstone of assurance is assessment of the small derivation steps by independent verifiers, possibly with variability and redundancy in verification methods. Traceability is the practical means to manage complexity along hierarchical decomposition paths. A by-product is diagnosability. In case of behavior deviation w.r.t. the specified, traceability-enabled backward dependence analysis enables precise localization of faults and errors. In turn, it enables fault elimination. Elimination of the known faults is characteristic of the correctness assurance regime. There would be no alternative to 100% accuracy in DAL A ML. Embedded known SPCF errors are ethically unacceptable.

ML violates the derivability and diagnosability assurance objectives of correctness policy. Approximant structure and parameter adjustment cannot be stepwise derived from training datasets. Consequently, when 100% accuracy is not reached, the root cause of fail-cases cannot be localized to enforce the error elimination policy. Correctness regime is intractable for ML, as of writing this paper.

D. No single point of failure

Regulation considers as unacceptable severe damage originating from a single specification, design, implementation, or operation error. Fault tolerant architectures are required. Since fault tolerance starts with fault detectors, on-line deviation oracles, in other words actionable specifications, are required. For ML, such actionable specifications are inaccessible on high-dimensional unstructured data like text, audio, and video signals.

E. *Correctness policy*

Software was regarded as a logical artefact that, in theory, could be developed without faults. By nature, it cannot spontaneously lose capabilities contrary to physical equipment. For these two reasons, standard committees applied fault prevention policy, i.e. correctness assurance to software. Safety standard committees regarded quantification of software reliability as ethically unacceptable for any safety-related development. In addition, it was deemed technically intractable in valid manner.

Like software, and contrary to physical equipment, ML model cannot spontaneously lose some capability as cause of a failure mode. They are deterministic, designed, time-invariant logical artefacts that make errors. Correctness regime should apply. However, miss of diagnosability prevents application of the "no-known-fault-left" policy.

F. *Rareness policy*

As seen previously, it could be an option for ML, considering the randomness sources in its elaboration process. However, it would be a paradigm shift to assimilate generalization errors to classical safety failure modes (i.e. random capacity losses). One would declare activation of preexisting flaws that are consequence of deliberate engineering choices, as equivalent to random physics-caused failures.

G. *Perspectives on SPCF-ML assurance*

The intent of the preceding review is to argue that there is no compelling choice of assurance policy for safety-critical machine learning. In addition, the application spectrum of ML is so large that a unique "one-size-fits-all" policy choice would be vain. Therefore, we reached consensus in our group on the following most flexible but principled rationale.

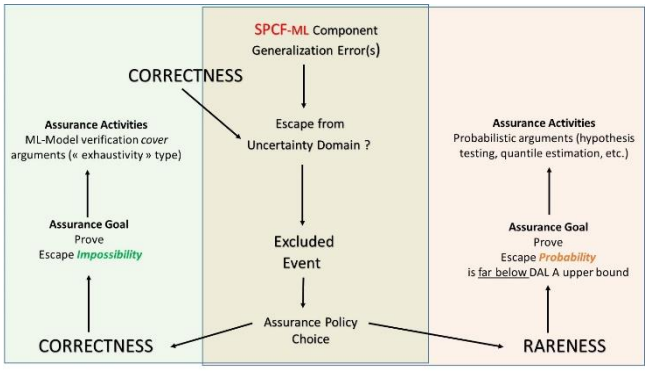

Fig. 4. Both options are sensible to some extent and missing means of compliance.

Since there is no compelling default option, our pragmatic stance is to leave the choice to the applicant, property-wise. For a given ML component, some failure modes could be assured by correctness means while others could be assured by probabilistic calculations. In our discussions, we even envisioned the case where a property could be partly demonstrated in correctness regime, and partly in rareness regime. Complexity of provably correct or ultra-reliable approximation in high dimension needs availability of any kind of well-founded verification technique.

VIII. COMPARING SW/1980S TO ML/2020S

Nearly half a century ago, software soared in embedded systems, while appearing brittle and raising concern about safety of software-intensive aircraft. In the early 80s, software-induced complexity ballooned as fast as grew the number of bugs per Kloc[8]. The foreseen "software crisis" for civil aviation lead to convention of assurance standard committees. First release of ED-12A/DO-178A was by 1982. About 40 years later, return on experience demonstrated that applying these assurance standards was effective.

ML and especially DL are following a similar trajectory: fast massive adoption by industry in spite of instable behavior (e.g. adversarial examples). Like for software, there are concerns about safety of ML-dependent aircraft or car. Automotive has been the leading industrial sector in the late 2010s. DL opened industrial viability of open- world computer vision. It made self-driving cars appear as a mid-term market opportunity. Consequently, development of ML-assurance standards started early, following ideas similar to that of proven-in-use software assurance standards. To what extent are these two histories comparable? Should we expect for ML assurance the success of software assurance?

A. *Similarities*

Foundations: a few decades before their respective booming industrial acceptance, both software and ML benefited from mathematical background: on computability and correctness for software (e.g. Turing, Floyd, Hoare); on statistical estimation, information and learnability for ML (e.g. Fisher, Shannon, Vapnik).

Engineering: in both cases these theoretical foundations had no immediate impact on tooling and industrial best practices.

High-dimensionality: software and machine learning share this characteristic. Curse of dimensionality to verify behavioral spaces is a common difficulty to meet the assurance requirements of the safety-critical. Safety-related embedded software has nowadays $D10^k$ input (resp. state, output) space dimensionality, with k possibly ranging from 2 to 7, and even beyond (e.g ATM/ATC ground segment software). It is the same dimensionality order of magnitude as that of DL-based HD video streaming processes.

Extensional verification cover: it was a deep problem for software assurance. One needed a sufficiency criterion to stop IVVQ activities with DAL-dependent appropriate confidence. Structural coverage, amenable to DAL modulation, was the solution. Committees were aware that even with MC/DC coverage, *extensionally* speaking, behavioral space cover was near zero. It was the best ALARP[9] cost/benefice trade-off at state of the art. Why then did software assurance succeed? Has near-zero extensional verification coverage the same significance for software as for machine learning?

B. *Disimilarities*

Point-based specification in high dimension and diagnostic inability seem to us the differentiating factors of

[8] Kilo-lines of code.
[9] As Low As Reasonable in Practice (risk)

ML w.r.t. software. Textual software specification are often example-based, i.e. scenario-based or use-case based. However, contrary to ML, all the examples are intended to be generalization seeds for human. Software developers generalize the examples when they formalize specifications and algorithms. Doing so, they implicitly create behavioral cells in their minds, named equivalence classes at testing stage. On the extensional side, these equivalence classes create volume-units of validity in the neighborhood of the 0-measure test cases. There is *implicit* augmentation of extensional coverage by principled code derivation and associated testing practice (i.e. requirement-based testing). Is there extensional coverage augmentation for ML, be it explicit or implicit?

IX. TDA-ENABLED (U)R-ML

We have justified why ML reliability must drastically improve to meet DAL A assurance objectives in the SPCF case. We have underlined a major difference between ML and software regarding verification cover: implicit volume-based cover for software, without any equivalent for ML. Foreseeable efficacy of ML assurance for the safety-related is likely to be far under the levels reached in the case of software.

We propose research orientations based on Computational Geometry (CG) and Topological Data Analysis (TDA) in higher dimensions [11], [28] to overcome these problems. It consists in supplementing classical statistical data science with awareness of topological complexity of datasets to support ML engineering activities like sampling, definition of In-Distribution oracles, diagnostic of inference errors, volume-based verification coverage analysis, empirical probability computation, etc.

In this section, we focus on sampling and explicit definition of the generalization domain (ID oracle). In the next and last section dedicated to the MNIST proof of concept, we adopt a broader view on use of topology.

A. Semantics of emptiness

High dimensional void is the ambient space around training and test point clouds. Emptiness around points may result either from principled choices, or from loopholes. Emptiness may be full of missing information that prevents from meeting correctness and/or reliability targets. We distinguish four types of voids:

1) Causal impossibility

Physics, scene or environment evolution laws, operational concepts or ODD constraints may prevent the generation of samples in definite regions of the input space. It leads to valid distant clusters or samples.

2) Sampling incompleteness

The sampling plan, compliant with the ODD and with the ML-model's textual specification, may overlook some input space regions. Depending on local regularity and approximant characteristics, these sampling lacunas may or may not constitute potential risk of inference errors.

3) Designed parcimony

When variability of data is under control, sampling may be appropriately parsimonious. Energy saving, or footprint constraints on embedded targets may also lead to local decimation of samples. In these cases, some extensive void regions are not risky.

For sampling coverage analysis, TDA should enable exploration of *dataset shape* to identify existence of unintended void regions (see fig. 5 and 10). It would consist in detection of non-interpreted large holes or cavities as potential sources of adjustment complexity and potential common modes for ensemble learning. This activity would be ML-model *independent* since it would only consider input spaces and ODDs as guide for data shape interpretation.

TDA offers a portfolio of algorithms to analyze point clouds in 2D, 3D, and in higher dimensions. We focus on persistence homology (PH) which plays a central role in TDA. It is used in ML for clustering, denoising, feature engineering (e.g. [12], [15]), and neural networks weight space or activation space analysis. We propose a new family of PH applications to machine learning whose overall goal is to overcome the reliability gap.

Roughly, PH computes a growing sequence of balls centered on each point of the dataset. For each ball radius of the sequential process named filtration, it computes the ball intersections and creates edges between the vertices that are centers of intersecting balls (see the four filtration steps of fig. 5). These edges constitute a nested mesh (simplicial complexes) that enables rigorous geometric and topological reasoning in higher dimension. They performs multi-scale modeling of point clouds. PH detects birth and death of kD-cycles, cavities and holes, as ball radius grows by discrete steps. It ends when the radius is so large that all balls intersect. Figure 5 illustrates some steps of 2D point cloud filtration.

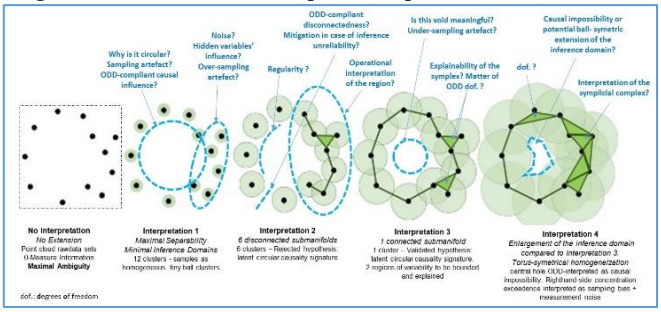

Fig. 5. Designing the inference domain, the "meaning" of the input part of training datasets. Four steps of persitence homology filtration are represented. In the upper part of the figure are examples of typical questions to interpret the filtration steps. At bottom we wrote examples of interpration decisions that could lead to selection of a given filtration parameter.

We propose to use PH filtration as (U)HR-ML data engineering practice to design some ODD-compliant interpretation of the training and testing datasets. Output of this task would be the ID-OoD[10] oracle of the approximant. For computational tractability, latent dimension must be far lower than ambient dimension.

B. Formal definition of inference domains

To our opinion, high reliability of approximants will require formal and executable definition of their domain (i.e. of their precondition from a formal method perspective). PH should offer means to define ID-OoD oracles in a way that does not depend on distributional assumptions or ML techniques [22].

[10] In Distribution – Out-of-Distribution

C. *Extensional verification coverage analysis*

We envision PH-based construction of a latent space simplicial complex as a means to guide scrutiny of generalization reliability. Triangulated training input spaces could support tight verification coverage criteria, simplex after simplex, used as generalization cells and as candidate counterpart of equivalence classes in software engineering. We name *extensional coverage* analysis this *volume*-based verification activity. It would be the extensional counterpart of structural coverage analysis in software. Such latent-space oriented verification coverage ideas are being explored for instance in [23].

D. *Contribution to ML safety assurance*

We first review four applications that are independent of any ML technique. This is a distinctive advantage since assurance values independence between design and verification methods.

1) Model-independent applications

1. Explicit generalization domains: using data augmentation, tuned filtration parameters, and PH simplices, design of a simplicial complex of operationally explainable generalization cells. The aim is an ID-oracle.
2. Designed separability: using persistence diagrams, homology groups, and homotopy classes as topological alerts of potential hard to fit regions for classifiers (cf. illustration on MNIST).
3. Extensional verification coverage analysis: using PH-complexes as *covers* of generalization domains, with multi-scale resolution.

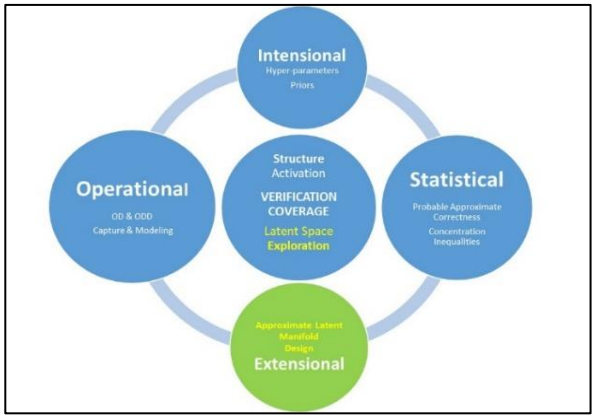

Fig. 6 Filling the ML reliability gap by enhanced verification coverage techniques. Extensional verification would ensure non-zero measure coverage, explicitly contrary to software where extension of equivalence classes remains implicit.

4. Novelty detection: non-stationarity tests in ML-Ops processes. TDA and Information Geometry could be used jointly to monitor datasets' shape trajectories and thoroughly diagnose risks of adjustment obsolescence.

2) Model-dependent

Research on how PH enables shape analysis of neural network activation spaces is undergoing. It has interesting potential for safety assurance as it could become in (U)R-ML engineering the extensional counterpart of structural coverage analysis and dead code elimination in safety critical software engineering.

X. PROOF OF CONCEPT ON MNIST

Last section is an outline of a proof of concept we are developing to support our discussions. It is also intended to support future (U)R-ML data-science challenges. The figures in this section do not result from TDA computation results, yet. They aim at presenting some (U)R-ML goals and activities, and at conveying intuition on a method whose engineering is still to develop. Preliminary results on digits {6, 0, 9} are documented in [33], to be made public after completion on the ten digits.

A. *Related work and discussion*

[31] is a systematic literature review devoted to certification of Machine Learning. Comparison with software is developed. There is no mention of the N-model non-independence problem. Topological data analysis is not mentioned either. [30] is another review of the main certification challenges for safety-critical ML. TDA is addressed and advocated as a promising approach. [28] is a survey of TDA applications to AI-ML, with focus on bio-molecular engineering. In image classification, all uses of PH reported in this survey are at image level, for dimensionality reduction, denoising, feature extraction, etc. In this PoC, we use PH at dataset level, to analyze the shape of the training and testing image databases.

In [26], PH applied to MNIST is reported. It enabled reducing 784D to 28D at iso-accuracy (96.3%). On our side, we want to augment accuracy (drastically), not to save computation time and energy without accuracy penalty. In [27], a table reviewing the performance scores of top10 MNIST classifiers is given. It provides evidence that reliability is plateauing at (1 - 3.10-3) on MNIST. We identified significant labelling errors in MNIST (~10-3 as well). It is a serious issue for (U)R-ML [41]. An ultra-reliable e-MNIST 10^7-sample dataset is needed (see fig. 8 for ambiguity cases of digits with letters).

In [34], persistence homology filtration of the testing dataset, and abstract interpretation of the neural network are combined. Goal is verification coverage analysis and global robustness verification. They adjust the filtration parameter to the ball radius used by the abstract interpreter. This work is the closest to ours in these last two sections. They use simplices for robustness cover only. We propose to use them also for explicit inference domains definition, and for *functional* property verification.

B. *Rationale of the Proof of Concept*

Our group is qualified to discuss safety assurance rationales. In the MNIST PoC, we adopt a safety assessment standpoint. As SPCF-ML is our focus, we consider the assurance objectives and activities of a team whose methods and tools should be independent of that in action by system and AI-ML development teams. TDA on raw datasets ensures independence w.r.t. statistical estimation.

TDA-enabled (U)R-ML is a sample-dependent method[11]. In this PoC, beyond independence w.r.t. statistical estimation, we also have independence w.r.t to ML-models. We concentrate on complexity of the problem to solve and regard topological complexity as a major risk of inference reliability.

[11] With confidence region of the ID triangulated manifold model

Topological Data Analysis on MNIST is applied to functional hazard analysis. Verifying stability of approximant behavior w.r.t dataset variability and optimization variability are non-functional risks. These assurance objectives address the engineering risks inherent to ill-posed inverse problems AI-ML is part of. They are of fundamental importance for ML life-cycle (e.g. MLOps), but they do not address correctness or rareness on functional failure modes.

Our PoC explores TDA support for verification of verification [12] : sampling coverage analysis and cross-validation coverage analysis.

C. Functional Hazard Analysis

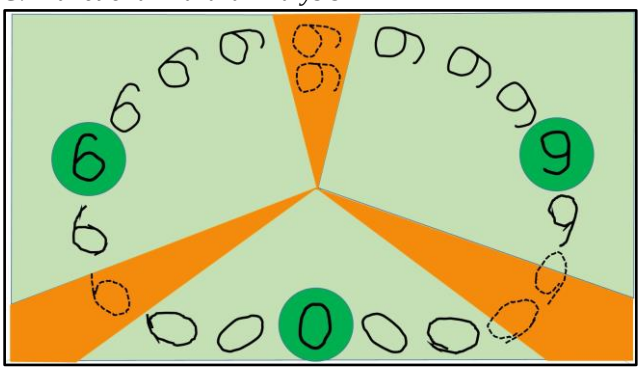

Fig. 7. Didactic eveocation of {6,0,9} homotopy equivalence, and of {rotation, translation, homotety} symetries. They create input-space hazardous regions (amber) subject to unreliable class separation by *any* ML-model. See fig. 10 for MNIST images belonging to hazardous regions (amber anulus).

LARD [25] is our planned next step in case of success on MNIST. We motivate our assurance activities by some fictitious aircraft landing narrative: we assume that some digits are painted on runways, and that their accurate recognition conditions safety-critical[13] operations.

1) Ambiguity

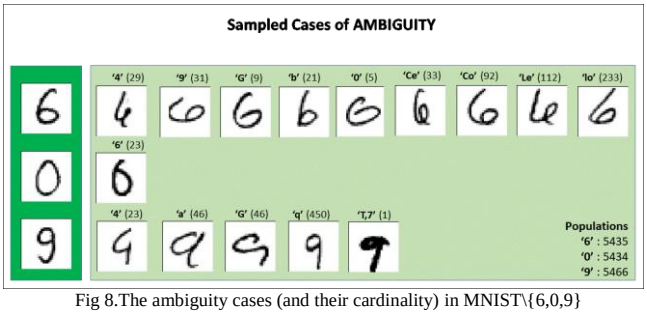

Fig 8.The ambiguity cases (and their cardinality) in MNIST\{6,0,9}

ODD of SPCF-ML MNIST classifiers should specify the image domain boundaries where even humans cannot decide. It should also identify where context-sensitive image interpretation occurs. For interpretation of distorted digits, knowing whether alphabetic characters may be present matters. (see 'a','le','co', etc. in fig. 8).

We follow and extend the perturbation taxonomy of [6]. We assume '0' has a distinctive operational role, and false positives on '0' are classified catastrophic by safety assessment, in the "no back-up" case. We assume false positives on '6' and '9' are hazardous. Detected false negatives on the three digits have no safety effect. Undetected false negatives are classified minor for safety, major for airport performance.

2) Perturbations

Light green background of the digits means that a single distortion (order 1) is regarded as common. Ultra-high reliability inference domains of {6, 0, 9}-classifiers should contain order 1 perturbed digits.

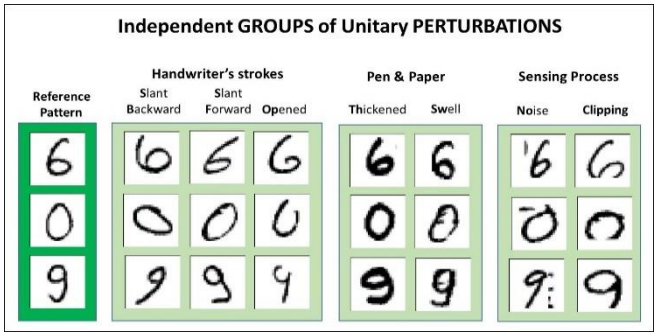

Fig. 9 Seven examples of "pure" perturbations (unitary, order 1), grouped by causal sources that are partially independent. Most of them combine freely up to high orders (e.g. a clipped+noisy+thickened+slanted digit is regarded as perturbed at order 4).

D. Safety objectives

We must ensure impossibility or extreme remoteness of False Positives on '0' ('0'-FPs). Choice of the assurance policy proposed to Authority is a critical issue. In both cases, 0-FP failure mode is an *excluded event* since we are in the SPCF case. If rareness assurance policy is chosen, probability of this failure mode should be demonstrated at one or two orders of magnitude below 10^{-9}/h. TDA in this PoC will be explored to support both policies.

E. Correctness policy

Current intermediary goal is a provably correct {6, 0, 9}-classifier on a restricted part of the inference domain (fig 11.left). Stratified persistence homology will be used to develop simplicial complex modeling of the inference domain over digits '0', '6' and '9' distorted by *unitary* perturbations. Unitary perturbations are the counterpart of component failure modes in classical safety (e.g. fault tree analysis). Their independence is an issue under rareness assurance policy.

A progressive *data-integration* process, counter-part of progressive *code-integration* process in software assurance, is enforced. PH is applied after every data integration step, to interpret growth of topological complexity, to locally augment data and to tune a subset of filtration parameters as multi-scale inference domain design decisions. Intuition of the data augmentation process is conveyed in figure 10. Separability on the ambiguity regions will be designed by simplicial engineering. Order 1 involves 3x7 local boundary designs (cf. fig 9), and 16 separability designs[14]. The resulting simplicial complex' actionable boundary will play the role of model-independent safety net.

[12] The assurance activities that verify that AI-ML verification activities are properly done.

[13] Admittedly, likelihood of runway hand-painted digits is extremely remote.

[14] "Separation" is somehow a misnomer. Most of the light green sub-clusters of fig. 11 share intersections. See the tessellation of sub-clusters as evocative of designed separation, or designed entanglement (e.g. fig1 like).

If not geometrically and combinatorically too complex, Order 2 will also be addressed under correctness policy (i.e. geometric models of decision boundaries and proofs by simplicial set inclusions or null intersections). Order 3 and beyond will be addressed only under rareness policy.

PH is necessary, but not sufficient for the envisioned (U)R-ML engineering. Implicit augmentation of the complex to address local and global symmetries is one of the needed additional ingredients.

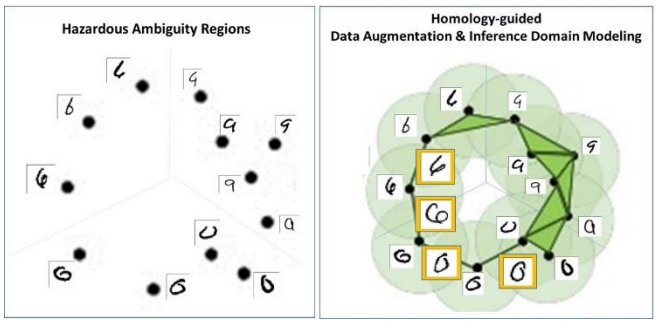

Fig. 10 Conceptual didactic figure derived from fig. 5. Left: a group of ambiguous distorted images. Right: for the selected radius of balls (filtration parameter – scale unit measure), PH seems to indicate the 12 images could be on a risky cycle like that of fig.7. Data augmentation (sampling or generation) is needed along the four newly created 1D simplices to confirm their relevance as new extensions of the inference domain..

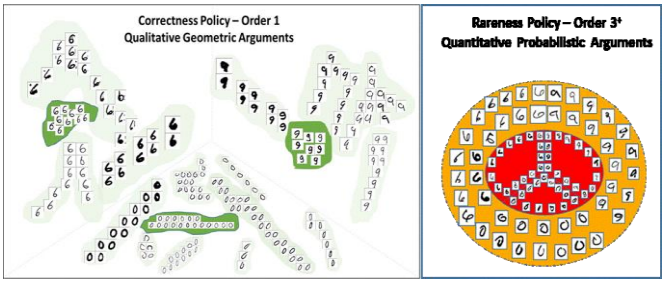

Fig. 11 Notional didactic figure sugesting the data-integration process. PH is a applied after every sub-cluster data increment, to assess the incrment of topological complexity.

F. Rareness policy

As order of perturbation increases, latent dimensionality and entanglement of shapes grow. Complexity of the ambiguity regions computed by PH become intractable for correctness policy. Conservative over-approximation of amber regions and probability estimation over triangulated manifolds will be the explored path. Its potential acceptance by certification bodies will be discussed in the group. Fault Tree Analysis should not be accepted as means of compliance to quantify failure modes in this context.

G. Current status and future work

MNIST restricted to {0, 6, 9} was sub-labeled to isolate 54 unitary perturbations (see 21/54 in fig. 9) and 16 ambiguity cases (fig. 8), as part of an independent safety assessment (U)R-ML process (Functional Hazard Analysis activity and safety requirements on cluster separability). PH was computed within 1 hour (~16500 images), and within 5 hours on whole MNIST (60000 images) on standard computing platform[15]. Interpretation of the persistence diagrams and design of the green and light-green ID[16]-boundary oracles are in progress.

Fig. 12 is the computed equivalent of fig. 11.left, limited to the three dense green regions (canonical ‘0’, ‘6’, ‘9’). Persistence. Three filtration steps (out of 50), growing from left to right like in fig. 5, are presented.

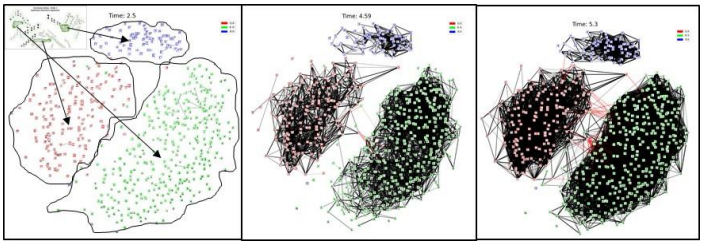

Fig. 12. Filtration step values are noted “time” (upper middle). The contacts (dashed red lines) appear at 4.59 between ‘0’ and ‘6’, and at 5.3 between ‘0’ and ‘9’ (ball radii).

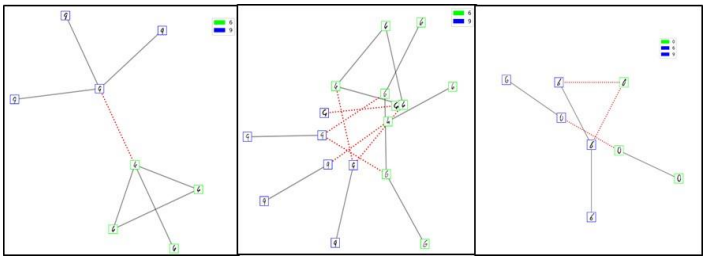

Fig. 13. Interpretation of ambiguity regions (contacts), when candidate connectivity for inference breaks class separation. PH with perturbated subclusters (fig 9).

XI. CONCLUSION

Starting from a reliability issue related to error correlation between AI-ML-model redundancies, we proposed a geometrical and topological explanation, not confirmed yet. We discussed the role of software development assurance and that of fault tolerant architectures to circumvent the problem. We argued that for ML components to be accepted as single points of catastrophic failure, like safety-critical software engineering and assurance managed to do, additional efforts and drastic progress on reliability are required.

We discussed the assurance regimes applicable to generalization errors in the most demanding case. We promoted a flexible approach and gave its underlying rationale. We proposed TDA as a candidate means of compliance to supplement statistical estimation theoretical guarantees. We limited ourselves to a safety assessment and ML-model independent perspective. We illustrated the envisioned methodology on a fictitious airborne SPCF MNIST classifier.

ML state of the art is progressing impressively fast. However, fundamental problems remain unsolved. We made explicit our top3 showstoppers: actionable specification, diagnostic inability, 0-measure specification and verification cover. We are confident that mathematics, algorithms and tooling maturation can fill the gap, as it was the case for software. We gave a first glimpse on TDA as a promising asset to substantiate this optimism. It will take time, as it was the case for software. New engineering has to emerge and mature, leaving many opportunities for applicants and Authorities to resist race to market expediencies.

ACKNOWLEDGEMENT

Our heartfelt thanks to Jean Paul Blanquart, emeritus safety expert at Airbus Defence & Space, for his pivotal role in assurance rationale analysis, and to Andrey Bychkov, PhD

[15] No use of accelerators at this stage.
[16] ID has an overloaded semantics: In Distribution and Inference Domain.

student at Thales Research, for his commitment in computing and visualizing PH clustering on {‘0’, ‘6’, ‘9’} of MNIST.

REFERENCES

[1] Lucian Alecu, Hugues Bonnin,Thomas Fel, Laurent Gardes, Sébastien Gerchinovitz, Ludovic Ponsolle, Franck Mamalet, Éric Jenn,Vincent Mussot, Cyril Cappi, & al. “Can we reconcile safety objectives with machine learning performances?“. ERTS2022, Jun2022, Toulouse, France.

[2] “EASA Concept Paper : First usable guidance for level 1 & 2 machine learning applications” March 2024. Issue 02.

[3] Morayo Adedjouma, Christophe Alix, Loic Cantat, Eric Jenn, Juliette Mattioli, et al.. “Engineering Dependable AI Systems”. 17th Annual System of Systems Engineering Conference (SOSE), IEEE, Jun 2022, Rochester, United States.

[4] Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Velickovic. “Geometric Deep Learning Grids, Groups, Graphs, Geodesics and Gauges. arXiv:2104.13478v2 [cs.LG] May 2021.

[5] Dan Hendriecks, Nicholas Carlini, John Schulman, Jacob Steinhardt. “Unsolved Problems in ML Safety“. arXiv:2109.13916v5 [cs.LG] 16 Jun 2022.

[6] Hermann Grassmann, “Extension Theory” 1862. History of mathematics Vol. 19. American Mathematical Society 2000.

[7] Jian Liang, Frederick Park, and Hongkai Zhao. “Robust and Efficient Implicit Surface Reconstruction for Point Clouds Based on Convexified Image Segmentation”. University of California, Irvine March 21st, 2011.

[8] Carlotta Giannelli, Lorenzo Sacco, Alessandra Sestini. “A local C2 Hermite interpolation scheme with PH quintic splines for 3D data streams”. arXiv:2108.12948v1 [math.NA] 30 Aug 2021.

[9] Clémentine Laurens, “Un vieux problème de courbes enfin bouclé”. Pour la Science N° 545, Mars 2023.

[10] Jean-Paul Blanquart, Philippe Baufreton, Jean-Louis Boulanger, Jean-Louis Camus, Cyrille Comar, Hervé Delseny, Jean Gassino, Emmanuel Ledinot, Philippe Quéré, Bertrand Ricque. “Software safety assessment and probabilities“. DSN 2016 Toulouse June 28th-July 1st.

[11] Jean-Daniel Boissonnat, Frédéric Chazal, Mariette Yvinec “Geometric and Topological Inference“ Cambridge Texts in Applied Mathematics 2018.

[12] Aditi S. Krishnapriyan1, Joseph Montoya, Maciej Haranczyk, Jens Hummelshøj, Dmitriy Morozov “Machine learning with persistent homology and chemical word embeddings improves prediction accuracy and interpretability in metal-organic frameworks” Nature Scientific Reports 11:8888 2021.

[13] Frédéric Barbaresco, Frank Nielsen Editors, “Geometric Structures of Statistical Physics, Information Geometry, and Learning. SPIGL’20, Les Houches, France, July 27-31.

[14] Marc Mézard “Désordre et frustration … et au-delà“ in Systèmes complexes, autour de Giorgio Parisi. Institut de France 11 octobre 2022 (unpublished communication).

[15] Mark Lexter D. De Lara, “Persistent homology classification algorithm“ PeerJ Computer Science January 10, 2023.

[16] Simon Martin, Pierre Yves Lagrave, “On the benefits of SO(3)-Equivariant Neural Networks for Spherical Image Processing. 2022. Hal-03763121.

[17] Herbert A. Simon “The Architecture of Complexity: Hierarchical Systems” in The Sciences of the Artificial, MIT Press 1969.

[18] Martin J. Wainwright ’High dimensional statistics – A Non-Asymptotic Viewpoint”. Cambridge Series in Statistical and Probabilistic Mathematics 2019.

[19] Kanti V. Mardia, Peter E. Jupp “Directional Statistics“ Wiley Series in Probability and Statistics 1999.

[20] Karim Benmeziane, Patrick Fabiani, Stephane Herbin, Jérôme Lacaille, Emmanuel Ledinot “Trusting Machine Learning Applications in Aeronautics” IEEE Aerospace Conference, Yellowstone, March 4-11 2023.

[21] Stephane Mallat, “Cours 3 : Malédiction de la grande dimension,” in L’apprentissage face à la malédiction de la grande dimension, Collège de France, 2018.

[22] Mohammadreza Salehi, Hossein Mizaei, Dan Hendrycs, Yixuan Li, Mohammad Hossein Rohban, Mohammad Sabokrou “A Unified Survey on Anomaly, Novelty, Open-Set and Out-of-Distribution Detection: Solutions and Future Challenges“ arXiv:2110.14051v1 26 oct. 2021.

[23] Taejoon Byun, Sanjai Rayadurgam “Manifolf for Machine Learning Assurance“ arXiv:2002.03147v1 8 Feb. 2020.

[24] Embedded France Groupe de travail - NSL Normes pour la Sûreté de fonctionnement Logiciel et système - Embedded France (embedded-france.org)

[25] LARD - Landing Approach Runway Detection – Dataset for Vision Based Landing - Mélanie Ducoffe, Maxime Carrere, Léo Féliers, Adrien Gauffriau, Vincent Mussot, Claire Pagetti, Thierry Sammour. HAL Id: hal-04056760

[26] Adélie Garin, Guillaume Tauzin “A Topological ‘Reading’ Lesson : Classification of MNIST using TDA”. aXiv 1910.08345v2 Oct 2019.

[27] Amarnath R, Vinay Kumar V, “Pruning Distorted Images in MNIST Handwritten Digits”. arXiv:2307.14343 May 2023.

[28] Chi Seng Pun, Kelin Xia, Si Xian Lee. “Persistent-Homology-based Machine Learning and its Applications: A Survey”. arXiv:1811.00252v1. Nov. 2018.

[29] Mary L. Cummings, Ben Bauchwitz. “Unreliable Pedestrian Detection and Driver Alerting in Intelligent Vehicles” IEEE Transactions on Intelligent Vehicles · January 2024.

[30] Alwyn Goodloe. “Assuring Safety-Critical Machine Learning-Enabled Systems: Challenges and Promise. Computer September 2023.

[31] Tambon, F., Laberge, G., An, L. et al. « How to certify machine learning based safety-critical systems? A systematic literature review”. Autom Softw Eng 29, 38 (2022). https://doi.org/10.1007/s10515-022-00337-x

[32] Bev Littlewood, “The use of Bernoulli and Poisson Processes for the evaluation of the reliability of critical software-based systems” Annex to IEC 61508 rev. 10, 2016.

[33] Andrey Bychkov, Emmanuel Ledinot. “(U)R-ML experiments on MNIST”. Technical Report, Thales Research & Technology, 2024.

[34] Faouzi Adjed, Mallek Mziou-Sallami, Frédéric Pelliccia, Mehdi Rezzoug, Lucas Schott, Christophe Bohn, Yesmina Jaafra. “Coupling algebraic topology theory, formal methods and safety requirements toward a new coverage metric for artificial intelligence models”. Neural Computing and Applications, 2022, 34 (19), pp.17129-17144.

[35] Patrick Grother, Kayee Hanaoka, “NIST Special Database 19 Handprinted Forms and Characters 2nd Edition”. August 2016.

[36] Daniel C. Castro, Jeremy Tan, Bernahrd Kainz, Ender Konukoglu, Ben Glocker “ Morpho-MNIST: Quantitative Assessment and Diagnostics for Representation Learning” Journal of Machine Learning Research 20 (2019) 1-29.

[37] JC Knight, NG Leveson, “An experimental evaluation of the assumption of independence in multiversion programming” IEEE Transactions on software engineering, 96-109.

[38] Júlio Mendonça1, Fumio Machida, Marcus Völp, “Enhancing the Reliability of Perception Systems using N-version Programming and Rejuvenation”, 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2023.

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, Rob Fergus, “Intriguing properties of neural networks”. arXiv:1312.6199v4 2014.

[40] Jabde, M., Patil, C., Mali, S., Vibhute, A. “Comparative Study of Machine Learning and Deep Learning Classifiers on Handwritten Numeral Recognition”, In: Thampi, S.M., Mukhopadhyay, J., Paprzycki, M., Li, KC. (eds) International Symposium on Intelligent Informatics. ISI 2022. Smart Innovation, Systems and Technologies, vol 333. Springer, Singapore.

[41] Curtis G. Northcutt, Lu Jiang, Isaac L. Chuang. “Confident Learning: Estimating Uncertainty in Dataset Labels” arXiv:1911.00068v6 [stat.ML] 22 Aug 2022.