# ML assurance in 6G-enabled edge-cloud continuum workflows

Marco Anisetti, Claudio A. Ardagna, Filippo Berto
*Computer Science Dept.*
*Università degli Studi di Milano*
Milan, Italy
{firstname.lastname}@unimi.it

Alex Della Bruna
*Consorzio Interuniversitario Nazionale per l'Informatica*
Rome, Italy
alex.dellabruna@consorzio-cini.it

***Abstract*—The modern edge-cloud continuum data intensive workflows are increasingly based on 6G edge nodes in order to spread their diffusion relying on public network and enhanced by the use of machine learning (ML) models in order to extend their capabilities. Data intensive workflows are also glowingly used in critical scenarios such as health and IoT. In these scenarios, guarantees on the model prediction quality and on the model non-functional properties (e.g., model confidentiality), are nowadays requested in order to comply with regulations such as the EU AI Act. Although the traditional CIA (Confidentiality, Integrity, Availability) triad are largely considered as the minimal non-functional properties to be guaranteed for a given system, they cannot be applied as such in the context of ML models. In this paper we identify the shortcomings of the conventional definition of CIA, provides novel ML-specific definitions for the CIA non-functional properties and develops an assurance methodology to evaluate them on the target models and provide relevant guarantees. The paper presents an experimental evaluation based on a realistic MLOps pipeline aimed to demonstrate its feasibility and effectiveness and is based on the novel definition of ML model integrity Non-Functional Property.**

***Index Terms*—ML, MLOps, Cloud, 6G, Edge, Assurance.**

## I. Introduction

ML models are becoming essential tools in an ever-wider range of applications, from healthcare to finance and autonomous systems. However, as these models are increasingly deployed in critical systems, stringent functional and non-functional requirements must be met. While functional requirements such as accuracy, precision, and recall are well studied in the literature, non-functional properties, such as security, privacy, integrity, and latency, are rarely considered [1] yet very crucial for maintaining both the system's trustworthiness and user confidence. Edge computing is currently emerging not just as the preferred way to address latency but also as a means to support trust. For instance, on-premises edge computing could reduce the need to share data and improve privacy through ML models. At the same time, edge computing can significantly reduce latency and bandwidth consumption by deploying ML models closer to data sources. The advent of 5/6G is enlarging the accessibility to edge computing in a transparent and reliable way by allowing users to exploit its potential directly without the need to set up specific edge nodes connecting them to the cloud. It also establishes a novel edge-to-cloud continuum by default at the telco networking level. However, such a continuum, while fundamental for supporting advanced applications based on ML models (i.e., by merging the advantages of edge and the computational capabilities of the cloud), is raising a number of challenges namely, the administration of deployment resources, the distribution of the application, its configurations and ML models, and the monitoring of applications functional and non-functional behavior in such heterogeneous environment. While some initial solutions exist to handle non-functional aware deployment in edge-cloud continuum [2]–[4], and to monitor the continuum non-functional posture [5], compliance to stringent regulations like the AI Act and GDPR in the continuum far from being addressed by the current literature especially when ML models is concerns. One of the key concerns is that the traditional non-functional properties of interest such as the CIA (Confidentiality, Integrity, Availability) that are needed to be addressed for compliance to the relevant regulations, cannot be applied as such to ML models. For instance, in the case of integrity applied to an ML model, it implies that the model cannot be modified after its release. In other words, this traditional integrity notion prevents the adoption of ML models in edge-cloud continuum scenarios where, in most of the cases, the model has to tune itself (e.g., by online partial re-training or tuning) to cope with the execution environment. In this paper, we propose a novel ML-specific interpretation of the CIA properties and an initial approach capable of evaluating them on a given ML model in the edge-cloud continuum. The rest of the paper is organized as follows: Section II introduces our notion of 5/6G empowered edge-cloud continuum and our

novel interpretation of the CIA non-functional properties of ML models deployed in such continuum. Section III describes a novel assurance methodology tailored to ML workflows in the continuum. Section IV presents our experimental evaluation focusing on integrity property only. Section V presents a discussion on efficacy and ease of integration in real-world ML workflows. Finally, Section VI summarizes the paper's contributions and discusses possible future advancements.

## II. The Scenario: ML-based workflows in Edge-cloud continuum

Nowadays ML models are increasingly available at the edge, for instance, to perform advanced analytics and fine-tuning on the data prior to be transferred to the cloud for further analytics. This scenario includes two specialized ML modeling workflows, the cloud-hosted training pipeline that generates an initial ML model to be transferred to the edge and the edge-hosted fine-tuning pipeline that specializes the model with local data. This scenario is emerging since it guarantees high performance of the cloud pipeline, thanks to specialized hardware, and improved data privacy on the edge. This kind of workflow is generally associated with MLOps processes that aim to guarantee the correctness, quality, and stability of the pipeline as its code and data change over time [6]–[8]. Although it is feasible to ensure advanced non-functional attributes throughout the entire ML workflows [9] and in the continuum-based deployment solutions [10], [11], this paper focuses primarily on the non-functional properties of the model itself. While the edge model is derived from an extended training dataset, it must respect relevant non-functional properties when moved to the edge. In this paper, we put forward the idea that the traditional CIA properties cannot be applied as such in this context but it is needed to revise them as follows:

- *Integrity*: the property of a model to demonstrate the integrity of its behavior across model updates. In our scenario the consistency of the edge model in relation to the base one in terms of their prediction behaviors (e.g., by showing the same or very similar predictions given the same or very similar input data).
- *Availability*: the property of a model to generate predictions within a reasonable time limit making the prediction usable in the context of the application. In our scenario the capability of guaranteeing a given prediction latency regardless of the edge platform (e.g., applying model pruning on an edge device to maintain the provided predictions requirements).
- *Confidentiality*: the property of a model to prevent the leakage of sensitive information. In our scenario, the leakage of sensitive information can be inferred from the model about the data used for the training (e.g., the possibility of generate/extract sensible data from the ML mode).

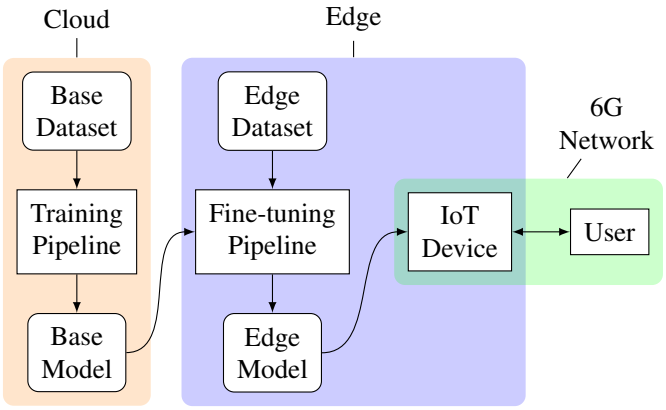


Fig. 1. Schema of our 6G-enabled edge-cloud continuum scenario.

Figure 1 shows our IoT edge scenario where Ml-based workflows are deployed in the continuum. We considered three areas: i) a cloud computing area utilized to train a classification model using a base dataset, thereby addressing the significant computational burden associated with computing on specialized hardware; ii) an edge area that fine-tunes the base model with a specialized dataset only available locally, and facilitates the distribution of the fine-tuned model to the IoT device(s). iii) A far-edge area in which an IoT device is responsible for handling classification requests from users who are connected through a local 5/6G network.

## III. The ML Assurance Methodology

In this section we describe our assurance methodology made of a framework and a process realizing the framework in the context of our scenario in Section II.

### A. Assurance Framework

Figure 2 shows the entire methodology based on the framework's components. It uses measurements collected by measuring the ML model state as objective evidence of its behavior. In order to evaluate a specific set of non-functional properties, an assurance verification process is established based on a set of contracts evaluating the collected measurements.

*1) Measurements:* are data points computed evaluating metrics in a time-series that can consider multiple aspects of the same target in the same time instant. For instance, we can apply a set of metrics to the trained model to measure its performance and monitor its behavior at each update.

*2) Properties:* describe intrinsic aspects of the application behavior. The paper focused on non-functional properties, as their validation is very relevant in the context of ML models, although very complex to achieve.

*3) Contracts:* are defined as formal specifications that outline the means of verifying a specific property on a given target. As illustrated in Equation 1, contracts can be defined as functions that map from a set of time-series data and a time instant to a Boolean value.

$$\mathbb{C} : \mathbf{S} \times \mathbb{T} \mapsto \mathbb{B} \tag{1}$$

Contracts are specific to the target platform, therefore their definition depends on the measurements available in the time-series.

**Example III.1.** Let us consider the availability non-functional property of a given ML model. It can be expressed as the time between a given prediction request and the relative model reply. Subsequently, a contract is established for the availability of the model. Let's assume that an acceptable reply time is one second, the relative contract can be defined as illustrated in Equation 2, wherein $latency$ serves as the metric for extracting the pertinent measurements.

$$\begin{aligned} c_{\text{model avail}}(t) = let \\ \mathbf{m}_{\text{latency}} = latency([t - 1min, t]) \\ in \max(m_{\text{latency}}) < 1s \end{aligned} \tag{2}$$

### B. Assurance Process

The assurance process is responsible for the verification of a subset of properties, resulting in the generation of a report that details the obtained results and the evidence utilized during the evaluation. The process is composed of three primary phases:

*1) Evidence mapping and collection:* In this phase, we verify which time series are necessary for a comprehensive assessment of the specified contracts in relation to the desired outcome. If not all the required measurements are available, the process will conclude prematurely with an error message. Contrary, the aforementioned measurements are collected locally for use in subsequent steps.

*2) Contracts evaluation:* In this phase the given contracts are evaluated using the collected measurements and each output is registered.

*3) Report generation:* In the last phase, we combine the evaluation outputs producing a report that contains both the output of each contract evaluation, and the relative evidence used to evaluate it.

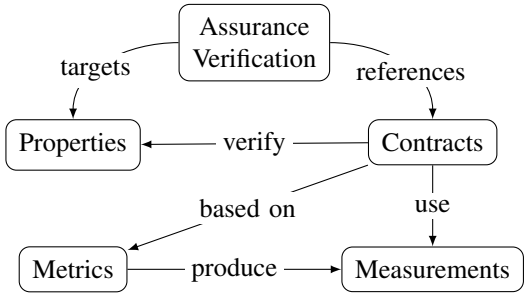


Fig. 2. Methodology components in the assurance process and their relationships.

## IV. Experimental Evaluation

In the following we first present our ML based case study in 6G-enabled edge-cloud continuum. We then experimentally evaluated the feasibility of our assurance methodology considering two distinct interpretation of the integrity property across model refinement/re-training: i) a traditional integrity notion defined as structural integrity of the ML model and ii) our novel integrity notion based on the integrity of the model behavior.

### A. Case Study and experimental setup

As a case study for our experimental evaluation we consider a IoT application that aims to classify objects in a camera video stream (i.e., a concrete example for the scenario in Section II). The IoT device is connected via 6G and uses an ML model deployed on telco edge to analyze the video, producing a list of the identified objects. The model is fine-tuned using data available only on the edge. This data are taken form the IoT camera thus reflect the peculiarity of the edge operation environment. The goal is to provide guarantees about the integrity of the *edge model* compared to the *base model* developed in the cloud, after the tuning occurred at the edge.

In this case study we consider as the *base model* "ResNet-50"[1], presented in [12], a deep neural network-based classification model pre-trained on the "ImageNet-1k" dataset[2] published in [13]. We derive a fine-tuned model by retraining only the last fully connected layer of the base model's neural network on the dataset "animals"[3] by applying Stochastic Gradient Descent (SGD). Specifically, the novel model is initially trained exclusively on a randomly selected subset of the training set, comprising half of the total examples.

[1]https://huggingface.co/microsoft/resnet-50
[2]https://huggingface.co/datasets/ILSVRC/imagenet-1k
[3]https://huggingface.co/datasets/mertcobanov/animals

Following this, process is repeated using the complete training set. This methodology enables the generation of two distinct ML models, the base one and the edge one, which are then subjected to comparative analysis to assess their integrity.

### B. ML Integrity verification

In the following we define our structural and behavioral integrity properties in terms of contracts to be used to evaluate them.

*1) Structural Integrity:* In order to ensure structural integrity, it is essential that the two models under comparison exhibit a comparable topology. Given that the weights of the two models have been fixed, except for the final fully connected layer, we may utilize the matrix Euclidean norm as a measure of distance, as demonstrated in Equation 3.

$$\|A - B\|_2 = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij} - b_{ij})^2} \tag{3}$$

The Euclidean norm possesses the distinctive property of consistently yielding a positive distance, with the resulting output being determined by a uniform distribution of contributions from all element-wise comparisons, while still considering the magnitude of their difference.

We can therefore define a contract for the structural integrity property as a threshold on the distance metric, as shown in Equation 4.

$$\begin{aligned} c_{\text{struct. integ.}}(M_1, M_2) = let \\ L_1 = M_1.\texttt{FC\_layers[-1]} \\ L_2 = M_2.\texttt{FC\_layers[-1]} \\ in\ \|L_1 - L_2\| \leqslant threshold \end{aligned} \tag{4}$$

*2) Behavioral Integrity:* The evaluation of the behavioral integrity is defined as the consistency between two models on a set of metrics with a given golden dataset. The golden dataset is defined on crucial data points where the behavior must be as similar as possible. If the two models demonstrate congruence in behavior on the golden dataset, it can be concluded that they are integral. In this example, the F1 score is considered as behavioral metric, both in terms of the entire dataset and in relation to each individual label. For F1 scores (i.e., label specific and overall) a threshold is defined in terms of the distance between the measurements, before which the two models are deemed to be integral. Equation 5 shows a formal definition of the behavioral integrity, with $S$ being the golden dataset and $S_i$ the part of the golden dataset set with the $i$-th label.

TABLE I
BEHAVIORAL INTEGRITY EXPERIMENTAL EVALUATION ON THE VALIDATION SET (THE FIRST 10 LABELS AND THE OVERALL SET).

| Label | F1 Score Distance |
|---|---|
| antelope | 0.127 |
| badger | 0.000 |
| bat | 0.457 |
| bear | 0.143 |
| bee | 0.000 |
| beetle | 0.190 |
| bison | 0.111 |
| boar | 0.179 |
| butterfly | 0.143 |
| cat | 0.111 |
| ... | |
| **Overall** | **0.0984** |

$$\begin{aligned} c_{\text{contx. integ.}}(M_1, M_2, S) = let \\ f1_{\text{tot}} = |F1(M_1, S) - F1(M_2, S)| \leqslant thr \\ f1_i = |F1(M_1, S_i) - F1(M_2, S_i)| \leqslant thr_i \\ in\ f1_{\text{tot}} \wedge \bigwedge_{i \in labels} f1_i \end{aligned} \tag{5}$$

### C. Experimental Results

In the following we compared the above behavioral and structural integrity contracts applied to our case study. In the case of structural integrity, selecting as distance threshold in the contract $8.00737$, the two models, the base and the edge one, exhibit substantial discrepancies in their weight distribution. Therefore with this threshold the structural integrity is not preserved. We note that with a threshold equal to $10$, the structural integrity contract is positive meaning that the integrity holds. It is clear that the threshold selection is crucial and have to be fine tuned.

In the case of behavioral integrity, Table I presents a subset of the results obtained on the validation set examples. The second column represents the distance between the F1 scores of the two models considering a given label and on the overall model. We note that a low value indicates a more coherent behavioral pattern. In our experiments we recorded values between $0$ and $0.600$, with an average distance of $0.124$ and a standard deviation of $0.145$. For the behavioral contract in Equation 5, we considered an overall threshold $thr$ of $0.2$ and, although $thr_i$ can be specific for each label $i$, for the sake of

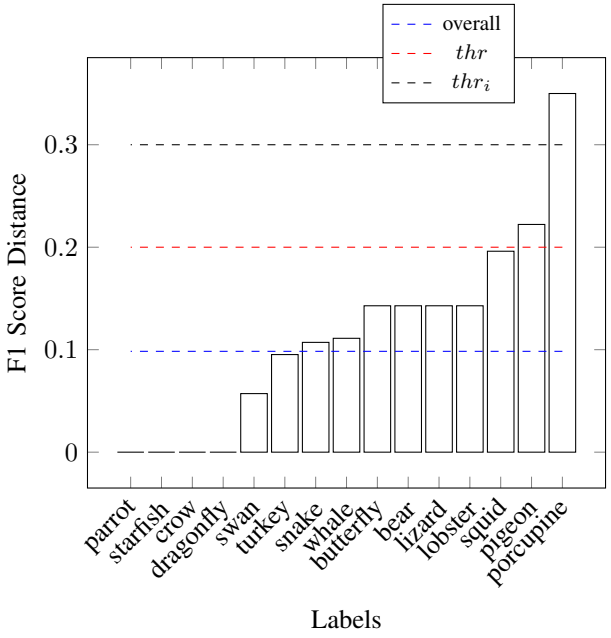


Fig. 3. F1 score distance of a sample comprising 15 labels compared with the overall, and the thresholds $thr$ and $thr_i$.

simplicity we considered the same threshold $thr_i = 0.3 \ \forall i$ as label-specific thresholds $thr_i$.

Figure 3 shows a random sample of 15 labels and the measured F1 score distance values, comparing them with the two thresholds $thr$ and $thr_i$. The figure shows that the overall threshold $thr = 0.2$ is higher than the computed overall F1 score ($0.0984$). On the contrary, the label-specific thresholds $thr_i = 0.3$ is violated by the label "porcupine", producing a negative evaluation for the entire contract defined in Equation 5. Similarly to the structural integrity case, the choice of thresholds is highly dependent on the specific domain and task at hand. Expert users must carefully consider these thresholds based on their knowledge of the data and the desired level of precision and recall.

## V. Discussion

Our experimental evaluation illustrated a case study where two versions of ML integrity properties are compared. The structural integrity definition employs a distance metric based on the weights of the two models to evaluate the similarity of their internal structures. The greater the similarity in internal structure, the greater the structural integrity of the two models. This approach is not without limitations. Its measure of integrity relies on the assumption that structurally similar models will behave consistently. While this may be true for relatively simple models, such as the one previously described, it may not accurately represent more complex models. Furthermore, the method is limited to models with analogous layer composition. If the number of weights in a layer differs, the Euclidean norm cannot be calculated. Moreover, if any of the labels are altered, the observed behavior may remain consistent, yet the underlying model structure may transform.

Behavioral integrity is a novel notion of integrity specific for ML models, where the focus shifts from the model's internal structure to its behavioral characteristics. The property definition is based on a comparison of the behavior of the models on the same golden dataset, with the objective of evaluating their consistency. This approach is more generalizable, as it can be applied to models with varying internal structures and is based on the model behavior itself, rather than an indirect structural measure. The golden dataset is fundamental and have to be defined in order to contain crucial data points where edge model integrity must hold. In other words the predictions of golden dataset datapoint of the base and edge models must not diverge significantly according to a given metric (in our experiments we uses F1 score). We note that our behavioral integrity is defined at different granularity level with a per-class and overall thresholds enables the definition of precise and tailored bounds. Ultimately, in both approaches the thresholds are to be determined by an expert, depending on the application context and the sensitivity required.

## VI. Conclusions

In this paper we described a novel assurance methodology for ML workflows in 6G-enabled edge-cloud for the CIA triad properties, showing how their standard definition do not map well with the nature of ML models, and providing new definitions tailored for this use case. We defined a representative scenario comprising a complete ML workflow, including cloud-based model training and on-edge fine-tuning. We then implemented our methodology on this scenario, thereby demonstrating the feasibility of two forms of integrity applied to ML models. Future works will include a complete framework of non-functional properties tailored to ML workflows, focusing on continuous verification of properties as part of MLOps processes.

## VII. Acknowledgments

This work is partly supported by the project MUSA – Multilayered Urban Sustainability Action – project, funded by the European Union – NextGenerationEU, under the National Recovery and Resilience Plan (NRRP) Mission 4 Component 2 Investment Line 1.5: Strengthening of research structures and creation of R&D "innovation ecosystems", set up of "territorial leaders in R&D" (CUP G43C22001370007, Code ECS00000037). It is also partially supported by Università

degli Studi di Milano via the program “piano sostegno alla ricerca” and “One Health Action Hub: University Task Force for the resilience of territorial ecosystems”, – PSR 2021 – GSA – Linea 6.

## REFERENCES


[1] M. Anisetti, C. A. Ardagna, N. Bena, and E. Damiani, “Rethinking certification for trustworthy machine learning-based applications,” *IEEE Internet Computing*, 2023.

[2] K. Fu, W. Zhang, Q. Chen, D. Zeng, and M. Guo, “Adaptive Resource Efficient Microservice Deployment in Cloud-Edge Continuum,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1825–1840, Aug. 2022, conference Name: IEEE Transactions on Parallel and Distributed Systems.

[3] M. Anisetti, F. Berto, and R. Bondaruc, “QoS-aware Deployment of Service Compositions in 5G-empowered Edge-Cloud Continuum,” in *2023 IEEE International Conference on Cloud Computing (CLOUD)*. IEEE, 2023, pp. 471–478.

[4] M. Anisetti, C. A. Ardagna, N. Bena, and R. Bondaruc, “Towards an Assurance Framework for Edge and IoT Systems,” in *2021 IEEE International Conference on Edge Computing (EDGE)*, Sep. 2021, pp. 41–43, iSSN: 2767-9918.

[5] F. Berto, C. A. Ardagna, M. Banzi, and M. Anisetti, “Assurance in advanced 5g edge continuum,” *IEEE Access*, vol. 12, pp. 178 659–178 671, 2024.

[6] S. Moreschini, D. Hästbacka, and D. Taibi, “MLOps Pipeline Development: The OSSARA Use Case,” in *Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems*, ser. RACS ’23. New York, NY, USA: Association for Computing Machinery, Aug. 2023, pp. 1–8.

[7] G. Recupito, F. Pecorelli, G. Catolino, S. Moreschini, D. D. Nucci, F. Palomba, and D. A. Tamburri, “A Multivocal Literature Review of MLOps Tools and Features,” in *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Aug. 2022, pp. 84–91.

[8] Y. Zhou, Y. Yu, and B. Ding, “Towards MLOps: A Case Study of ML Pipeline Platform,” in *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, Oct. 2020, pp. 494–500.

[9] M. Anisetti, C. A. Ardagna, and F. Berto, “An assurance process for Big Data trustworthiness,” *Future Generation Computer Systems*, vol. 146, pp. 34–46, Sep. 2023.

[10] M. Anisetti, F. Berto, and M. Banzi, “Orchestration of data-intensive pipeline in 5G-enabled Edge Continuum,” in *2022 IEEE World Congress on Services (SERVICES)*. Terassa, Spain: IEEE, Jul. 2022, pp. 2–10, iSSN: 2642-939X.

[11] M. Anisetti, N. Bena, F. Berto, and G. Jeon, “A DevSecOps-based Assurance Process for Big Data Analytics,” in *2022 IEEE International Conference on Web Services (ICWS)*. Barcelona, Spain: IEEE, Jul. 2022, pp. 1–10.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.