AVALIA

AI Competitions and Benchmarks Platform

Proposal and Technical Sheet

Adrien Pavão, Ihsan Ullah

Contents

1 Context

1.1 The importance of competitions and benchmarks for AI

Artificial Intelligence (AI) is an empirical science. Progress comes from running experiments, measuring outcomes, and refining methods based on the results. As AI systems are increasingly deployed in society, ensuring that these results are trustworthy and reproducible has become essential. This requires experiments to be conducted in controlled, open environments where the research community can verify and build upon them.

To determine which methods actually solve a given problem, such as image classification or sentiment analysis, they must be compared under the same conditions. This is the role of benchmarks and competitions: to define a problem rigorously and evaluate all proposed solutions according to identical rules [Pavão, 2023]. A **benchmark** is a standardized task with a fixed dataset and

evaluation protocol, allowing methods to be compared on equal footing. A **competition** builds on this format by inviting participants to submit solutions within a defined time frame, typically with results ranked on a public leaderboard.

Three major benefits justify their use. First, competitions **mobilize collective intelligence**: by opening a problem to many participants with diverse profiles, organizers obtain more varied and often better-performing solutions than what a single internal team would have imagined. It is one of the few known ways to surface genuinely new approaches to complex problems, in a cost-effective manner. Second, the format **avoids the inventor–evaluator bias** that arises when a team designs both the method and its evaluation. A competition strictly separates the two roles: the organizer defines the problem and evaluates submissions fairly, under identical rules for every participant. Third, competitions **produce measured evidence**. Without a benchmark framework, performance announcements remain unverifiable; a competition yields a reproducible, comparable, and publishable measurement, which is essential when deciding whether to put a system into production.

1.2 Existing platforms and their limitations

Existing *Benchmarks* and *Competitions* platforms, such as Kaggle, CodaLab, Codabench, and Zindi etc. have instrumentalized the "challenge" format to accelerate AI development. However, despite their success in fostering community engagement, they leave several fundamental needs unanswered, particularly regarding scientific rigor and industrial constraints.

The limitations of current systems can be categorized into three main areas:

- **Methodological fragility:** Most contemporary leaderboards prioritize raw performance scores over statistical significance. The absence of confidence intervals and little control over the baseline often leads to "leaderboard climbing", where marginal gains are indistinguishable from random noise. Furthermore, the lack of built-in protections against test-set exhaustion (repeated submissions) undermines the validity of the reported results.
- **Constraints on sensitive data:** Current benchmarking architectures typically require data to be hosted on a central server. This introduces limitations of usage for sectors governed by strict privacy regulations such as healthcare, finance, or defense. Consequently, these fields remain largely excluded from the collaborative benefits of open benchmarking, as they cannot risk the exposure of sensitive datasets.
- **Narrow evaluation metrics:** There is a significant disconnect between "leaderboard performance" and "production readiness". Existing platforms focus almost exclusively on the usual evaluation metrics, e.g. accuracy, F1-score, RMSE, etc., ignoring critical operational dimensions such as computational efficiency, inference latency, and energy consumption. This one-dimensional evaluation fails to identify solutions that are truly viable for real-world deployment.

1.3 Proposed framework: *AVALIA*

We introduce ***AVALIA***, a next-generation benchmarking and competition platform designed to move beyond the limitations of current systems. Rather than simply hosting leaderboards, *AVALIA* provides a robust, extensible framework that prioritizes scientific integrity and the operational viability of submitted solutions.

The platform is built around a modular architecture that is easy to integrate into existing research workflows while maintaining the rigor required for high-stakes AI evaluation. Specifically, our framework is designed to directly mitigate the three critical failures identified in the current ecosystem:

- **Rigorous methodology:** Rather than presenting a static list of scores, *AVALIA* treats benchmarking as a statistical experiment. The platform enforces methodological best practices directly in the infrastructure: principled ranking functions, hidden evaluation sets to avoid overfitting, baseline comparisons, and uncertainty quantification through bootstrap resampling. Rankings therefore reflect genuine differences in performance rather than statistical noise.
- **Handling sensitive data:** We introduce a new architecture for accessing data and evaluating submissions that securely store data in a decentralized way and allow restricted access to the compute resources to use the data securely and evaluate submissions. By leveraging containerized environments, the platform allows competition organizers to evaluate third-party models on local, private servers without the sensitive raw data ever leaving the host's infrastructure. This opens the door for high-stakes collaborative research in medical imaging, education, defense, and other sensitive domains.
- **Evaluating real-world applicability:** *AVALIA* expands the definition of "performance" to include the hidden costs of AI. Every submission is profiled across a multi-dimensional metric space that accounts for inference latency, peak memory usage, and estimated carbon footprint. By surfacing these trade-offs, the platform identifies models that are not just accurate on paper, but sustainable and efficient for real-world deployment.

Our objective is to develop *AVALIA* in France as a new cutting-edge research software, with a deliberate European positioning. This is not incidental: it ensures alignment with European data protection regulations (notably GDPR), supports digital sovereignty for sensitive applications in healthcare, defense, and public research, and connects the platform to the academic and industrial actors of the region. Rather than building on top of existing systems, the platform is designed from scratch so that the methodological and architectural principles described above can be embedded directly into its core.

2 *AVALIA* – Introduction to the platform

2.1 Vision and positioning

Solving sensitive, high-impact problems

The project targets a clear niche: organizations of significant size (industrial actors, research institutes, sensitive bodies) that have specific problems, confidential data, and a real need to compare several algorithms on a business problem.

Typical target profiles:

- Industrial actors (for example, an automotive manufacturer) wishing to organize multiple internal benchmarks on various problems: electronic optimization, material handling, predictive maintenance of materials and vehicles.
- Research institutes in sensitive domains such as education, health, and energy etc., that struggle to adopt AI because their data is too confidential. They install the system once and then run multiple public or internal benchmarks (imaging, automatic diagnosis, etc.).
- Bodies such as National Deefense Organizations: license purchase with maintenance, with no visibility on our side regarding internally organized benchmarks.

The major weakness of current platforms is the failure to measure the **real-world practicality** of solutions, and the lack of a framework adapted to **sensitive data**. The exploratory or playful focus (small hackathons, theoretical research on "who will get the best score") is not our target.

We address stakeholders that need AI for sensitive subjects and want to benefit from a rigorous methodology and benchmarking system.

2.2 Business model

We primarily sell software under license, optionally accompanied by consulting services to help install a complete platform and organize benchmarks, competitions, and hackathons. The core value proposition is the **methodological pipeline and protocol** for selecting a good solution to a given problem, as well as the rigorous formulation of the problem itself. The distinctive feature of the software lies in its **methodological** and **security** strength.

Below we propose multiple ways in which the platform can be used:

- **Restricted Licensed Use** *(Exact terms to be defined)*: The software is provided under a *source-available* model, where the code is fully readable for transparency and auditing but remains proprietary. Usage is strictly governed by a restrictive license that requires formal permission or a paid agreement for deployment. No redistribution allowed unless prior permission is granted.
- **Consulting packages:** We offer specialized consulting packages to assist with the professional installation of the platform and technical configuration and management of individual benchmarks and competitions.
- **Managed Benchmarking Infrastructure:** We provide a dedicated environment to host public competitions and benchmarks for clients with standard security requirements. This platform includes baseline compute support and the flexibility for organizers to integrate their own external compute resources for more demanding workloads.

2.3 Platform architecture

Central principle

The platform architecture decouples the orchestration layer from the execution and storage layers. This separation keeps the datasets and code within the client's secured environment while a lightweight public hub manages coordination. This architecture is illustrated in Figure 1.

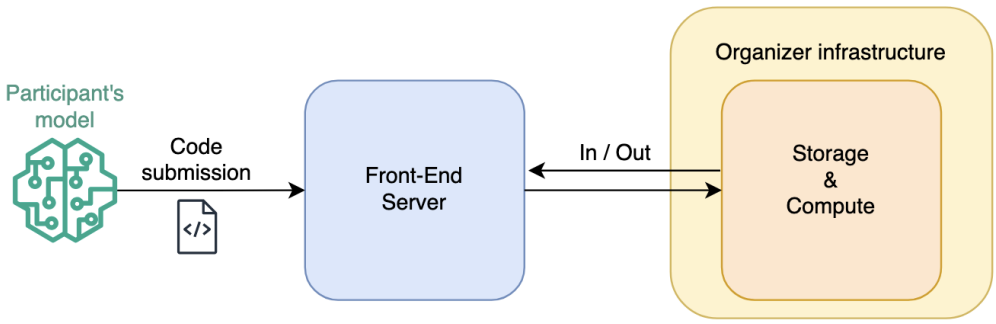

Figure 1: General architecture of the system. Our host service coordinates submissions and displays sanitized results. Sensitive data and computations remain within the client's infrastructure.

Architectural components

The system is a modular, containerized ecosystem with four primary layers:

- **Frontend:** A React-based web portal for administrators, organizers, and participants, covering profile, team, competition, and submission management.
- **Backend:** A Python service handling core logic, API coordination, the database, user permissions, the submission queue, and the connection to compute resources.
- **Compute:** Evaluation tasks run in isolated Docker containers, with support for attaching external compute resources for heavier ones.
- **Data storage:** Only public metadata and sanitized scores are kept on the host hub; datasets, ground truth, and submission code reside on the client's local storage or S3-compatible buckets, accessed only at evaluation time.

Architectural benefits

The decentralized design addresses several critical challenges inherent in traditional benchmarking platforms, particularly regarding data sovereignty and operational overhead. The following points outline the primary problems mitigated by this architecture:

- **Data sovereignty and residency:** Datasets and ground truth remain at all times on the client's premises. This eliminates the legal and security risks associated with transferring sensitive information to third-party servers.
- **Intellectual property protection:** Beyond the raw data, the client maintains exclusive possession of the benchmark's internal logic and source code.
- **Operational cost efficiency:** By leveraging the client's existing compute and storage infrastructure, the platform avoids the high costs associated with maintaining a massive, centralized execution environment. This allows for a more scalable and sustainable model for both the provider and the client.
- **Full administrative autonomy:** The client retains total control over the lifecycle of their benchmarks. This includes the ability to delete, restructure, or modify assets at any time without external constraints or synchronization delays with a central authority.
- **Zero-trust security model:** Access to the datasets is restricted to the specific execution window of a submission. Because the platform infrastructure only handles sanitized metadata and final scores, it creates a physical and logical barrier that prevents unauthorized access to the underlying sensitive data.

Three possible levels of security

There are three possible security levels for deploying the platform (Figure 2):

1. All components hosted in our infrastructure, which is not preferred for sensitive use cases.
2. Front-end and Back-end hosted on our side, while storage and compute remain in the organizer's infrastructure.
3. All components deployed within the client or organizer's infrastructure, offering the highest level of privacy.

Level (2) is particularly interesting: the hybrid architecture is where the platform brings the most distinctive added value, allowing organizers to keep data and compute fully on-premise while sharing only the front-end and back-end. Level (3) also has value for organizations with strict isolation requirements, offering them a robust and convenient benchmarking system, though it corresponds to a more classical fully on-premise deployment.

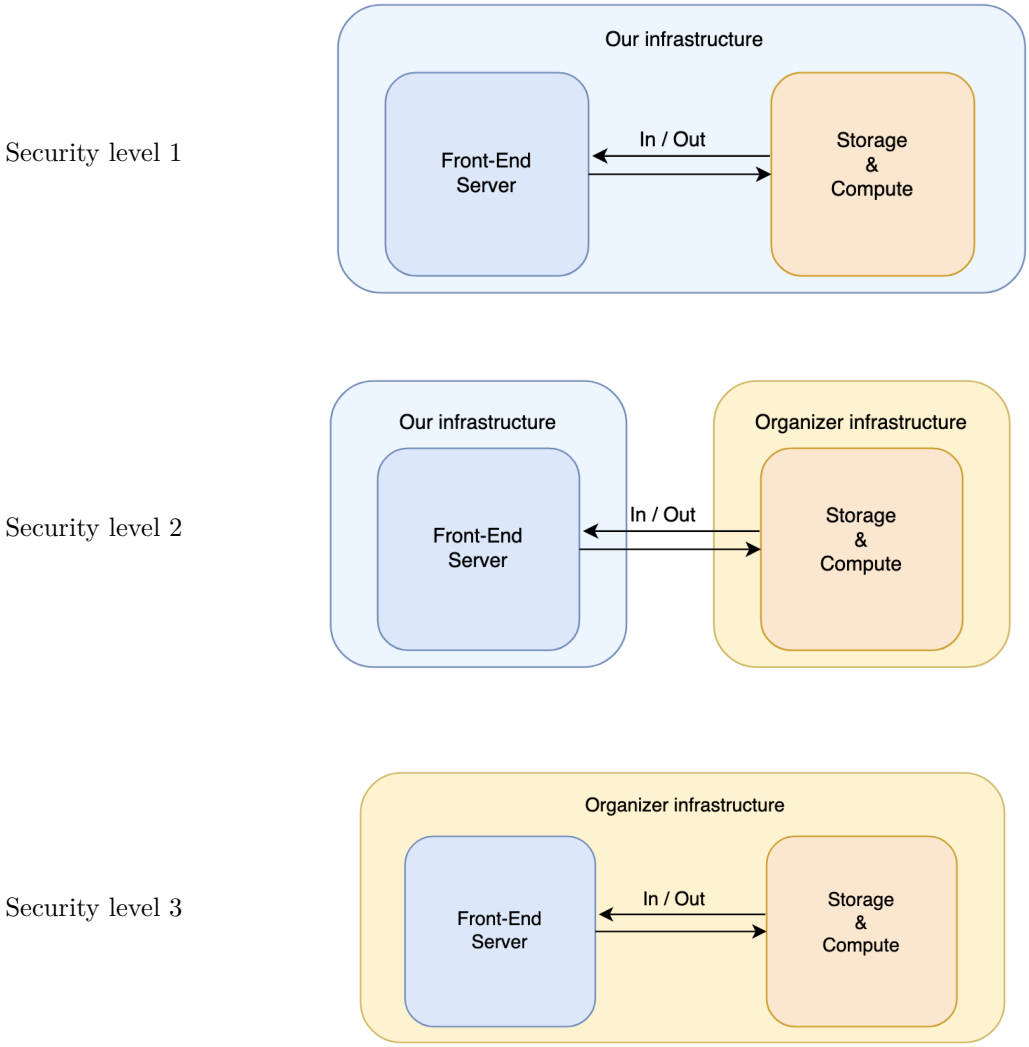

Figure 2: Three possible security levels for deploying the platform: (1) all components hosted in our infrastructure (2) front-end hosted on our side, while storage and compute remain in the organizer's infrastructure; and (3) all components deployed within the client or organizer's infrastructure

2.4 Benchmark methodology

The platform implements rigorous scoring and validation protocols so that rankings reflect statistical reality rather than stochastic noise. By embedding these methodologies directly into the infrastructure, we enforce best practices that are often difficult to apply manually.

Ranking functions

A benchmark rarely reduces to a single metric: a submission may be judged on predictive performance, computational cost, and other criteria that are not directly comparable. The platform handles this heterogeneity natively by aggregating multiple scores into a single ranking. Rather than relying on ad-hoc score combinations, we use the *average rank* method, a simple yet robust approach whose stability across datasets and resistance to outliers are well supported by our research [Pavão et al., 2021a, Pavão, 2025]. When organizers want to align the leaderboard with specific research or industrial priorities, a *weighted average rank* variant lets them place

more emphasis on selected criteria.

Public and private leaderboards

A core principle of rigorous benchmarking is the separation between a public evaluation set, used during development, and a hidden private set, used to assess generalization. The platform supports this separation natively, with multiple public and private leaderboards per phase so that organizers can group related metrics together. Phases themselves can be configured as sequential, following the classical "development then final" workflow, or parallel, with several leaderboards running side by side over the same period.

This dual-set design matters even when the benchmark is purely internal. The organizer rarely cares only about raw performance: they want to inspect submissions and select the solution that fits their constraints. A "final phase" in the public sense has little meaning here, but the methodology still applies. A second private evaluation set, accessible only to the organizer, is sufficient to verify that submitted solutions generalize. The private score can either be recomputed at each submission, when compute allows, or evaluated at a fixed deadline; in both cases, the people trying to solve the problem do not have access to it. Far from being optional, this discipline is what makes internal evaluation trustworthy, and we see it as a selling point of the platform.

Visibility controls are fully configurable. The "private problem, public participation" scenario is particularly important: the organizer retains full visibility over all submissions, while participants only see restricted or anonymized feedback. This is essential when the underlying data or task is sensitive.

Statistical significance

The performance of a submission is never an exact value: it depends on the specific evaluation sample. The platform systematically provides error bars computed through bootstrap resampling, so that the gap between two submissions can be interpreted in terms of significance rather than raw score difference. Bootstrapping gives good balance between statistical solidity and computing cost [Pavão et al., 2025a, Pavão, 2025]. Submissions whose performances are statistically equivalent are visually grouped on the leaderboard, preventing the premature ranking of solutions whose differences lie within statistical noise.

Baseline as a threshold

Each benchmark includes a baseline, or reference method, clearly defined by the organizer and visible on the leaderboard. Organizers can require that a submission outperform this baseline by a statistically significant margin to be considered a valid solution [Pavão et al., 2022]. This filters out "leaderboard climbing" on marginal, non-reproducible gains. It also has a direct implication for the business model of an open competition: the organizer is only liable for a prize when a participant produces a solution that meaningfully outperforms the baseline.

2.5 Real-world practicality and operational utility

Beyond simple ranking, benchmarking serves as a collaborative tool for identifying industrially viable solutions.

Performance beyond accuracy

A model that achieves state-of-the-art accuracy on paper but cannot run within a target latency, memory budget, or energy envelope is of little use in production. Yet most public benchmarks

still report raw performance as the only metric, leaving these operational dimensions invisible. We take the opposite stance: a benchmark is only useful when it surfaces the trade-offs that determine whether a solution can actually be deployed.

Every submission is therefore profiled along several axes alongside predictive performance. **Execution** time is measured for both training and inference, giving a direct view of computational cost. Memory and CPU usage are tracked per pipeline stage rather than only in aggregate, which makes it possible to locate bottlenecks instead of merely reporting them. **Energy consumption** is recorded as well, providing an estimate of the environmental and hardware footprint of each solution. These measurements are not optional add-ons: they are integrated into the ranking itself, so that solutions which balance classical performance metrics with operational efficiency are surfaced over those that win on a single dimension at unacceptable cost.

Pipeline evaluation

This per-stage measurement is made possible by the platform's modular pipeline architecture. Each evaluation is structured as a sequence of well-defined stages, such as data preprocessing, model training, and inference, that the platform executes and instruments independently. Because the stages are explicit, the platform adapts naturally to diverse problem types: zero-shot tasks omit the training phase, evaluation-only benchmarks skip preprocessing, and custom workflows can introduce additional stages without breaking the evaluation pipeline. The same structure produces fine-grained telemetry, exposing execution time and resource consumption at the level of individual components rather than as opaque totals, and it lets organizers configure the leaderboard to reflect the operational requirements that matter for their use case, for instance by weighting inference cost more heavily than training cost when latency is the binding constraint.

2.6 Code versioning and environment management

Source control and development workflow

The platform leverages **integrated Git repositories** (such as GitHub or GitLab) on the organizer's side to streamline benchmark development. This integration allows engineers to manage training and evaluation logic using standard version control workflows, significantly reducing the overhead required to initialize or update complex competitions.

Containerization and reproducibility

To ensure a **strictly identical environment** for all participants, the platform utilizes containerization. This approach guarantees:

- **Fairness:** Every submission is executed with the exact same dependencies, libraries, and hardware configurations.
- **Reproducibility:** Evaluation results remain consistent and verifiable over time.
- **Sustainability:** Competitions are easier to maintain and archive, as the specific compute environment is preserved within the container image.

While the system supports custom environments, it prioritizes existing image registries (such as DockerHub or private GitLab repositories) to avoid redundant infrastructure.

Automated re-evaluation

To prevent discrepancies caused by updates to the training/evaluation code or container images, the system supports automated recomputation. When an organizer updates a task, the platform can automatically rerun all existing submissions so that the leaderboard reflects the current

environment. Since compute resources are dedicated to each client, these updates can be performed without impacting other users or incurring shared costs. When automatic re-evaluation is disabled, the platform flags outdated results to prevent confusion and ensure accurate interpretation of the leaderboard.

2.7 Full control over submissions

Code submissions

The platform is built around **code submissions** rather than result-only submissions, because code is what makes the rest of the framework possible. Submitting code, rather than predictions, means the organizer can keep the evaluation data hidden from participants: the model is brought to the data instead of the data being shipped to the participant. It also enables direct measurement of the computational footprint of each submission (Section 2.5), which is impossible when only output files are uploaded. Finally, code is replicable: any submission can be re-executed under controlled conditions to verify the reported results. Result-only and hybrid modes remain available for benchmarks where this is the natural format, such as classical prediction tasks, but the platform's distinguishing capabilities are designed around code submissions.

The submission workflow then takes each entry through a sequence of stages, illustrated in Figure 3: input, pre-execution verification, decentralized storage, and execution in an isolated environment.

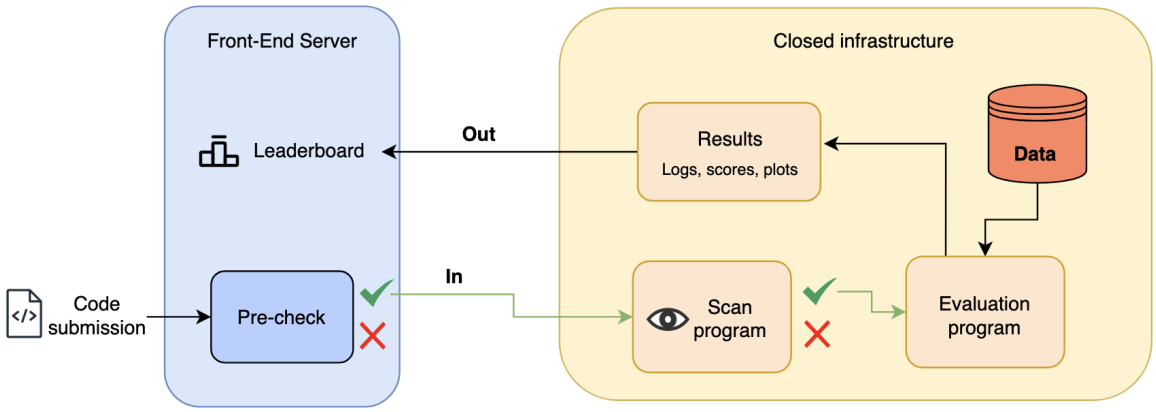

Figure 3: Detailed architecture of the submission pipeline: input code submission, pre-execution verification (human or automatic), decentralized storage, and execution in an isolated environment.

Pre-execution verification

When a participant uploads a submission, it is immediately validated against organizer-defined constraints: allowed file types and sizes, expected entry points, and rate limits or per-participant daily quotas to control both compute costs and the risk of test-set exploitation. The submission proceeds further only if these checks succeed ("Pre-check" step in Figure 3).

Before any submission reaches the compute resources, organizers can require it to pass another verification ("Scan program" step in Figure 3). Two complementary mechanisms are available: **automated scanning** inspect submissions for malicious patterns, formatting issues, or policy violations; organizers can rely on predefined scan templates, plug in their own scanner code, or, for advanced cases, use a locally-running LLM to flag suspicious code without ever exposing it to an external service. For sensitive or high-stakes benchmarks, a **human-in-the-loop protocol** can

also be enabled, in which a submission remains pending until an authorized organizer manually reviews and approves it. Both checks are optional and can be combined: a submission only proceeds to execution once the configured checks return a successful status.

Execution environment

Submissions are executed inside dedicated containers. By default, the container has no network access, which guarantees two important properties at once: a strong security boundary, since code cannot exfiltrate data or call external APIs, and reliable resource measurements, since CPU, GPU, and memory usage are not perturbed by network latency or external computation. Only sanitized performance metrics are returned to the platform; the submitted code itself remains inside the compute environment.

For benchmarks that legitimately need network access, such as evaluations involving API-based LLMs, organizers can explicitly enable it. In that case, the platform flags the benchmark so that participants and observers know that compute time and memory measurements are no longer strictly comparable across submissions, since part of the work may run on external services. To preserve fairness, the platform always pulls the same container image and evaluation logic from the linked Git repository, so every submission is processed under identical hardware and software constraints, regardless of network configuration.

Decentralized submission storage

To maintain data sovereignty, the platform employs a decoupled storage strategy. While the platform database stores necessary metadata (e.g., timestamps, participant IDs, submission ID, submission name, and status, etc.), the actual submission files are never stored on the central platform infrastructure. Instead, they are routed directly to the **linked storage** configured for the competition, which may be a cloud-based bucket (AWS S3, GCS, Azure Blob) or local storage within the client's private hardware.

3 Who are we?

3.1 MLChallenges

MLChallenges[1] is French consulting company for AI Competitions and Benchmarks, founded by Adrien Pavão and Ihsan Ullah. It provides diverse services for organizing competitions and benchmarks, setting up a complete competition platform, and handling scientific work related to competitions.

MLChallenges has scientific and technical experience with challenges organization and has organized several large-scale competitions for Lawrence Berkeley National Laboratory, Université Paris-Saclay, University of Washington, Institute for Advanced Study, ChaLearn, U.S. Department of Energy, Rensselaer Polytechnic Institute, among others. Our commitment is to make crowdsourced challenges a powerful scientific tool.

We are the core developers of the competition platform *Codabench* [Xu et al., 2022], as illustrated in Figure 4, and have shaped the platform, improving its functioning and adding features related to transparency, reproducibility and results communication.

We describe two examples from our work below:

[1] https://mlchallenges.com/

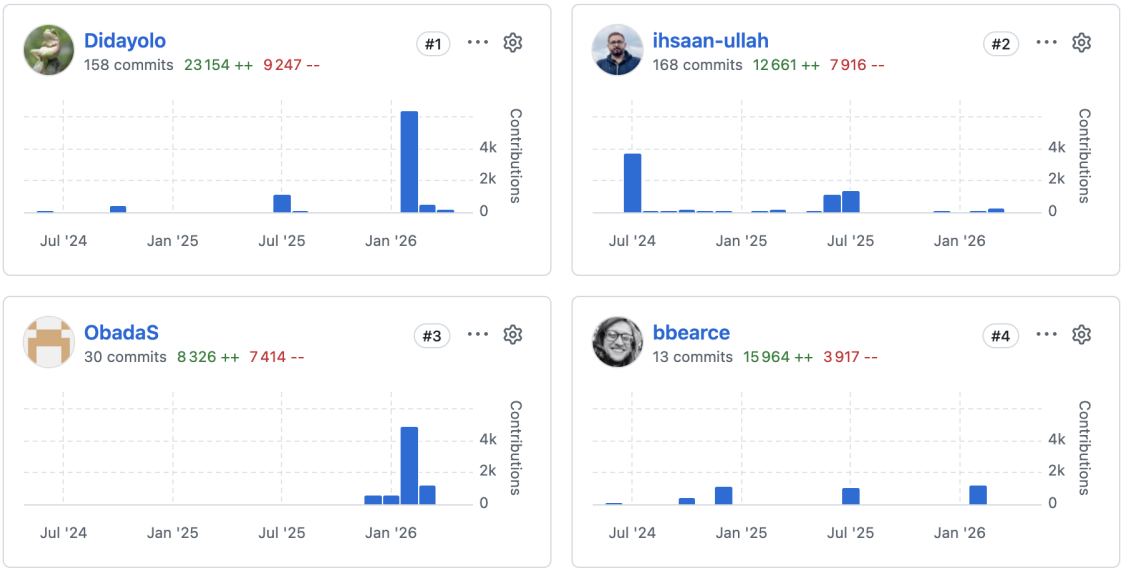

Figure 4: Contributions from Adrien (Didayolo) and Ihsan (ihsaan-ullah) to Codabench's Github repository.

Paris Region Data Challenges

The Ile-de-France (Paris Region) organized data challenges jointly with French industries, and the LISN[2] laboratory from Université Paris-Saclay, to encourage innovation and scientific progress on high-level technical problems.

In this context, we were directly involved in the design, technical setup, and analysis of the "*Dassault-Aviation - AI 4 Industry Challenge*" [Pavão et al., 2021b] and "*Learning to Run a Power Network Challenge 2023 (L2RPN)*" [Pavão et al., 2025b]. Both competitions involved a €500,000 research grant award for the winning team, and a robust scientific protocol for evaluating participants' submissions.

Dassault-Aviation challenge's task was a multivariate time series regression on aeronautics data. Each submission took around 24 hours to compute on dedicated GPU hardware, as it involved code submission and on-server training of the models. We developed a protocol to be able to work with confidential data, on a blind setup where participants trained their models on data they were not able to directly access.

In **L2RPN 2023**, organized with Réseau de Transport d'Électricité (RTE), participants provided AI agents capable of operating a simulated power grid, inside a virtual environment. This example shows how Codabench, jointly with our competition organization expertise, can handle diverse evaluation protocols, such as reinforcement learning environments.

FAIR Universe - Uncertainty-aware large-compute-scale AI platform for high energy physics and cosmology

The FAIR Universe[3] project [Benato et al., 2025, Dai et al., 2026] is a collaboration between scientists from Lawrence Berkeley National Laboratory, Université Paris-Saclay, University of Washington, Institute for Advanced Study, and ChaLearn. The collaboration is building a large-compute-scale AI platform for the hosting of large scientific datasets, models, and machine learning competitions to drive forward discoveries in high-energy physics and cosmology. The

[2] https://lisn.upsaclay.fr/
[3] https://fair-universe.lbl.gov/

initial focus is on discovering and minimizing systematic uncertainties in High Energy Physics through a sequence of challenges that will bring together researchers across physics and machine learning.

On the technical side, we set up a dedicated private instance of Codabench for internal use at the National Energy Research Scientific Computing Center (NERSC), configured both CPU and GPU compute workers and connected them to the private Codabench instance. Additionally, many bugs were fixed in the open-source Codabench project and several new features were added that were beneficial for these competitions and overall Codabench users. Multiple GitHub repositories were managed for the competitions that include the competition code, FAIR Universe website, synthetic data generation code and more. This project is almost concluded and was a successful project that has contributed well to the scientific community of High Energy Physics and Machine Learning.

3.2 Adrien Pavão

Adrien is PhD in machine learning from Université Paris-Saclay, specializing in the methodology of scientific competitions [Pavão, 2023]. Lead editor and co-author of the reference volume *AI Competitions and Benchmarks — The Science Behind the Contests* [Pavão et al., 2025a]. He has organized numerous scientific challenges in collaboration with academic research labs and industry partners. He is also a core developer and administrator of the competition platforms *CodaLab Competitions* [Pavao et al., 2023] and *Codabench*.

3.3 Ihsan Ullah

Ihsan is a Research Software Engineer and a core contributor to the Codabench platform. He graduated from Université Paris-Saclay with a master's degree in Artificial Intelligence. His expertise spans Machine Learning and Software Engineering, specifically large-scale research infrastructure. Ihsan has organized several ML Competitions and brings more than 3 years of expertise in scientific challenges (design, evaluation, and management) and research software and infrastructure.

3.4 Why we are in a position to lead this project

We combine three elements rarely brought together in a single team: **leading academic research** on benchmark methodology, **operational experience** of organizing dozens of international competitions, and **technical mastery** of the existing platforms (CodaLab, Codabench), of which we know both the strengths and the limits. We have a strong experience in software engineering to take on such ambitious project. This dual position, as researchers and platform builders, gives us a precise vision of what is currently missing for industrial and public actors who want to rigorously evaluate AI solutions.

References

Lisa Benato, Wahid Bhimji, Paolo Calafiura, Ragansu Chakkappai, Po-Wen Chang, Yuan-Tang Chou, Sascha Diefenbacher, Jordan Dudley, Ibrahim Elsharkawy, Steven Farrell, Aishik Ghosh, Cristina Giordano, Isabelle Guyon, Chris Harris, Yota Hashizume, Shih-Chieh Hsu, Elham E. Khoda, Claudius Krause, Ang Li, Benjamin Nachman, Peter Nugent, David Rousseau, Robert Schoefbeck, Maryam Shooshtari, Dennis Schwarz, Benjamin Thorne, Ihsan Ullah, Daohan Wang, and Yulei Zhang. Fair universe higgsml uncertainty dataset and competition, 2025. URL https://arxiv.org/abs/2410.02867.

Biwei Dai, Po-Wen Chang, Wahid Bhimji, Paolo Calafiura, Ragansu Chakkappai, Yuan-Tang Chou, Sascha Diefenbacher, Jordan Dudley, Ibrahim Elsharkawy, Steven Farrell, Isabelle Guyon, Chris Harris, Elham E Khoda, Benjamin Nachman, David Rousseau, Uroš Seljak, Ihsan Ullah, and Yulei Zhang. Fair universe weak lensing ml uncertainty challenge: Handling uncertainties and distribution shifts for precision cosmology, 2026. URL https://arxiv.org/abs/2604.14451.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6, 2023. URL http://jmlr.org/papers/v24/21-1436.html.

Adrien Pavão. *Methodology for Design and Analysis of Machine Learning Competitions*. PhD thesis, 2023. URL https://theses.fr/2023UPASG088. Thèse de doctorat dirigée par Guyon, Isabelle Informatique université Paris-Saclay 2023.

Adrien Pavão. *How to judge a competition: Fairly judging a competition or assessing benchmark results*, chapter 4. Submitted at DMLR, 2025. URL https://ai-competitions-book.github.io/chapters/chapter4.pdf.

Adrien Pavão, Michael Vaccaro, and Isabelle Guyon. Judging competitions and benchmarks: a candidate election approach. *European Symposium on Artificial Neural Networks (ESANN) Proceedings*, 2021a.

Adrien Pavão, Zhengying Liu, and Isabelle Guyon. Filtering participants improves generalization in competitions and benchmarks. In *30th European Symposium on Artificial Neural Networks*, 2022. URL https://www.esann.org/sites/default/files/proceedings/2022/ES2022-72.pdf.

Adrien Pavão, Isabelle Guyon, Evelyne Viegas, et al. *AI Competitions and Benchmarks – The Science Behind the Contests*. Submitted at DMLR, 2025a. URL https://ai-competitions-book.github.io.

Adrien Pavão, Antoine Marot, Jules Sintes, Viktor Eriksson Möllerstedt, Laure Crochepierre, Karim Chaouache, Benjamin Donnot, Van Tuan Dang, and Isabelle Guyon. Ai challenge for safe and low carbon power grid operation. *Energy and AI*, 22:100564, 2025b. ISSN 2666-5468. doi: https://doi.org/10.1016/j.egyai.2025.100564. URL https://www.sciencedirect.com/science/article/pii/S2666546825000965.

Adrien Pavão et al. Airplane numerical twin: A time series regression competition. *International Conference on Machine Learning and Applications (ICMLA)*, 2021b.

Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7):100543, 2022. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2022.100543. URL https://www.sciencedirect.com/science/article/pii/S2666389922001465.