SIEVE: Towards Verifiable Certification for Code-datasets

Fatou Ndiaye MBODJI University of Luxembourg Luxembourg, Luxembourg fatou.mbodji@uni.lu

> Kui Liu Huawei China kui.liu@huawei.com

El-hacen Diallo University of Luxembourg Luxembourg, Luxembourg el-hacen.diallo@uni.lu

Jacques KLEIN
University of Luxembourg
Luxembourg, China, Luxembourg
jacques.klein@uni.lu

Jordan SAMHI
University of Luxembourg
Luxembourg
jordan.samhi@uni.lu

Tegawendé F. BISSYANDE University of Luxembourg Luxembourg, Luxembourg tegawende.bissyande@uni.lu

Abstract

Code agents and empirical software engineering rely on public code datasets, yet these datasets lack *verifiable* quality guarantees. Static "dataset cards" inform, but they are neither auditable nor do they offer statistical guarantees, making it difficult to attest to dataset quality. Teams build isolated, ad-hoc cleaning pipelines. This fragments effort and raises cost. We present SIEVE, a community-driven framework. It turns per-property checks into *Confidence Cards*—machine-readable, verifiable certificates with *anytime-valid* statistical bounds. We outline a research plan to bring SIEVE to maturity, replacing narrative cards with *anytime-verifiable* certification. This shift is expected to lower quality-assurance costs and increase trust in code-datasets.

CCS Concepts

· Software and its engineering;

Keywords

dataset certification, confidence sequences, code datasets, software engineering datasets, reproducibility, continuous auditing, community-driven validation, machine-verifiable evidence

ACM Reference Format:

1 Introduction

Data underpins modern science and machine learning. It powers recommendation systems, code-generation tools, and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, Washington, DC, USA

 \odot 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM https://doi.org/10.1145/nnnnnn.nnnnnnn products used at global scale. Yet dataset trust remains fragile: once published, we often cannot tell if a dataset is complete, clean, or legally compliant. If a dataset contains biases or compliance failures, the flaws propagate, compromising research validity and seeding failures in deployed systems. Other domains (e.g., chip design, infrastructure) certify quality before use. However, for datasets, the foundation of empirical science, we still lack transparent, machine-verifiable certification.

Early documentation efforts set the norm for human-readable records: Datasheets for Datasets formalized a structured questionnaire covering motivation, collection, and limitations [1]; the Data Nutrition Label proposed modular summaries to surface issues at a glance [2]; and Data Cards emphasized user-centric, purpose-driven documentation to aid responsible deployment [3]. To bridge prose and pipelines, recent work standardizes machine-readable metadata: Open Datasheets contributes a JSON schema to export structured documentation that downstream systems can parse [4]; Croissant-RAI define a Web-native vocabulary for lifecycle, labeling, safety/fairness, and compliance, enabling direct load and validation of RAI (Responsible AI) metadata [5]. While these efforts standardize RAI integration, their effectiveness depends entirely on adoption by dataset providers.

In reality, dataset documents remain scarce. An audit of 7,433 Hugging Face dataset cards found that only 30.9% of repositories contain non-empty cards, although those datasets account for 95% of downloads [6]. Even among the most popular datasets, the critical section "Considerations for Using the Data" which should describe biases, limitations, and downstream impacts averages only about 2.1% of the content [6]. At the same time, the EU AI Act requires providers to publish training-data summaries and maintain technical documentation for regulatory oversight [7]. The gap between regulatory expectations and current practice illustrates how far the ecosystem is from evidence-backed dataset certification.

Beyond under-documentation, risks are already materializing: widely adopted datasets may carry biases or violations, yet they have been used to support scientific conclusions. [8] shows massive indirect leakage of benchmark data into closed-source LLMs during evaluation.

Code-datasets particularly differ from other corpora: they are executable artefacts whose auditing is both operationally and semantically demanding. In practice, audits require

Property ¹	CodeNet [9]	CSNet [10]	HumanEval [11]	APPS [12]	The Stack v2 [13]
Buildability	Yes	No	No	No	No
Test smoke	Partial	Partial	Partial	No	No
Link valid	No	No	N/A	No	No
Dependency health	No	No	No	No	Partial
License resolves	No	Yes	No	No	Partial

Evidence pointers:

(Yes)

- \bullet CodeNet build ability in the "status" column;
- CSNet: licenses for the source code in the _licenses.pkl

(Partial)

- CodeNet: tests provided but Only for AIZU;
- CSNet: human relevance judgement are given;
- HumanEval: function to test generated in the "test" column;
- The Stack v2 : acknowledge that the training dataset could contain malicious code and the limitation of license attribution

(N/A)

HumanEval: APPS card: data are handwritten.

(No)

• Informations not found in the dataset cards

Figure 1: Documentation coverage of practitioner–critical properties as advertised in dataset cards/docs. Yes = explicitly documented as addressed; $\frac{Partial}{Partial}$ = partially/indirectly stated; $\frac{Partial}{Partial}$ = not stated; $\frac{Partial}{Partial}$ = not applicable.

reconstructing toolchains, pinning compilers and package registries, resolving transitive dependencies, and running builds/tests whose outcomes can drift as ecosystems evolve. Meanwhile, repositories become inaccessible, APIs deprecate, new CVEs surface, and stale projects silently bias analyses—making "the same dataset" hard to reproduce across time and machines.

To better understand real needs, we conducted a survey (Cf. 2) from which we identified recurring properties required by code datasets. Figure 1 contrasts what popular code-ataset cards currently document with these needs.

Gap in Datasets and Objectives

Gap. While the ecosystem is converging on standards for where information should reside (Croissant-RAI), and regulators are demanding more (EU AI Act), to the best of our knowledge, there is no measurable evidence on the quality of code datasets, and even less concerning the properties demanded by researchers and practitioners. Objective. SIEVE: the pioneering solution toward a transparent, machine-verifiable, per-property certificate for code datasets, reporting quality with anytime-valid statistical bounds. These certificates provide verifiable proof of dataset quality.

2 Understanding Dataset Challenges

This section investigates practical challenges encountered when using code datasets.

Interview

We conducted semi-structured [14] interviews. The details are given in the table 1.

Recruitment	Participants contacted with study overview
Format	online or face-to-face
Participants	18: (15 SE researchers and 3 AI engineers)
Focus	Dataset quality challenges

Table 1: Interview methodology summary

-#	Interview	Onection

- 1 What common quality challenges have you encountered in code datasets?
- ? How have you identified concerns or issues in datasets you worked with?
- 3 What suggestions do you have for improving dataset documentation and reporting of issues
- Can you provide examples of specific datasets where such issues were observed?

Table 2: Key questions asked during the interviews.

Key Findings:

Table 3: Key interview insights on code-dataset issues

g .	Aspect	Observation		
	Indirect discovery (compliance)	Compliance risks (licensing) are rarely detected directly; they surface via colleagues, talks, or reviews.		
	Missed or low-quality capture	Valuable data is often not captured; indiscriminate scraping and weak filters yield noisy or low-quality corpora.		
•	Abandonment pattern	Teams frequently invest time, then abandon datasets due to quality issues; many cannot later recall the dataset names.		
-	Recall shaped by feedback	Datasets criticized by reviewers or reused by peers are more salient than those abandoned quietly.		

As summarized in table 3, our interviews with SE researchers and practitioners surfaced three recurring patterns: (i) dataset issues are rarely reported and projects are often quietly abandoned, wasting effort; (ii) compliance and policy risks (e.g., licensing, sensitive content) are typically discovered indirectly and late in the workflow; and (iii) even within the same group, teams in different SE subareas do not share signals, so common risks remain invisible. In short, quality problems are discovered reactively rather than proactively. These observations motivate our approach: replace ad-hoc, one-off cleaning with a proactive, certification layer. Accordingly, we are designing a systematic analysis of widely used code datasets to identify concrete manifestations of these issues and to prioritize the property definitions and pinned oracles that SIEVE will certify.

Informed by the insights from these interviews and targeting a potential solution, below, we present our proposal: SIEVE.

¹Properties definitions: buildability = repo builds in a smoke run; test_smoke = if tests exist, a short run passes; link_valid = entries resolve to repo+commit; dependency_health = vulnerable dependencies; license_resolves = license present & compatible.

3 Proposed Framework: SIEVE

As datasets gain value, public and private stakeholders invest heavily in cleaning and maintaining ever-changing corpora. They need continuous, reproducible assurance of quality, yet current efforts are fragmented and often duplicate the same dataset pre-process work. SIEVE empowers the stakeholder consortium to co-sponsor datasets and collaboratively refine their quality and properties on an ongoing basis. It also transforms checks into transparent, machine-verifiable certificates with quantitative guarantees, thereby reducing redundant effort and enhancing trust.

3.1 Global View

Actors and roles. As depicted in Fig.2a, sponsors submit datasets for audit. They also bear the cost of processing the entire audit and provide rewards as incentives for validators. The reward is assumed to be a recognition asset, similar to academic contributions such as reviewing papers. In scenarios involving private entities, continuous submission of their local test datasets for validation may be directly enforced by the sponsors. Sponsorship-related business models fall outside the scope of this work.

Because reviewing datasets containing hundreds of millions of records is both complex and expensive, sponsors may not require a full row-by-row assessment. Therefore, we introduce two tolerance measures per property: (i) an error bound ε , which specifies the accepted error on a given property, and (ii) a coverage parameter $1-\delta$, which limits the cost derived from auditing.

Validators are dataset users (e.g., researchers, engineers) who derive the public samples and run lightweight property checks (oracles) on these samples. Through sponsors, validators may also define properties aligned with their needs.

Arbiters reproduce validator evidence, aggregate results, and attest the current confidence score. Their role can be configured differently depending on the deployment. In an academic context, arbiters may act as reviewers who simply aggregate and recheck validators' claims; in other settings, AI models could serve this role. In all cases, arbiters are auditable, and validators may challenge their outputs. If a conflict arises, a contradiction report is issued to highlight violations of the attestation.

Smart Contract. SIEVE leverages a contract ² as a trust anchor that makes a dataset audit transparent and verifiable for stakeholders. It anchors the dataset and audit rules, fixes public randomness for unbiased sampling, escrows and settles funds under transparent rules, and keeps an append-only log of attestations and challenges. All checks run as off-chain evidences; the chain stores only commitments, ensuring independence from the sponsor and reproducible audits with an on-chain footprint.

3.2 Confidence Card

A Confidence Card is a machine-readable record stating, for a dataset version and a binary property P (violation/noviolation), the current evidence: sample count t, observed violations S_t , a live interval L_t , U_t for the true violation rate p, and a decision state. It is updated as more items are checked and can be replayed by any third party. We use anytime-valid confidence sequences (CS): at every sample count t (number of distinct items evaluated), CS provide an interval for p that remains valid no matter when we look or stop (continuous monitoring).

Assumptions. Uniform seeded sampling. deterministic, version-pinned oracle; tolerance ε and coverage $1-\delta$ fixed.

Guarantee. We maintain L_t, U_t such that

$$\Pr(\forall t \ge 1: p \in L_t, U_t) \ge 1 - \delta,$$

valid under arbitrary peeking/stopping.

Construction (Bernoulli, KL time-uniform). Let

$$da \parallel b = a \log\left(\frac{a}{b}\right) 1 - a \log\left(\frac{1-a}{1-b}\right),$$

denote the binary Kullback–Leibler divergence between Bernoulli parameters a and b, and define the anytime penalty

$$\psi_t \delta = \log \left(\frac{2 \log_2 2t}{\delta} \right),$$

as in [15–17].

At each time t, we invoke a standard routine that maps $t, \widehat{p}_t = S_t t, \delta$ to a confidence interval L_t, U_t using a time-uniform Bernoulli bound (we adopt the KL-based formulation of [15]). Specifically,

$$\begin{cases} U_t = \inf\{ u \in \widehat{p}_t, 1 : t \, d(\widehat{p}_t \parallel u) \geq \psi_t \delta \}, \\ L_t = \sup\{ \ell \in 0, \widehat{p}_t : t \, d(\widehat{p}_t \parallel \ell) \geq \psi_t \delta \}. \end{cases}$$

3.3 Workflow

This section presents the SIEVE workflow, which is structured into the following steps and illustrated in Fig. 2b:

- (1) The sponsor submits a dataset for audit including:
 - DatasetID = rootHash, URLs : exact dataset version (e.g., commit SHA/CID) and eventual link to the dataset.
 - Property set $\mathcal{P} = \{P_j, \varepsilon_j, \delta_j\}_{j=1}^J$
 - Oracles: content digests (e.g., repo+commit) of the checker for each P_j .
- (2) The contract rejects duplicates for the same rootHash and locks a public randomness seed. All parties derive the same uniform schedule of indices via a pseudorandom function.
- (3) Repeated until a terminal decision:
 - (a) Validators submit the next unclaimed seeded index; arbiters enforce membership and de-dup.
 - (b) For each property P_j , compute $X^j \in \{0, 1\}$ on the sampled item.
 - (c) Publish {indices, bits, oracles, logs} to a offchain store (e.g., IPFS) and its digest/URI on-chain.

 $^{^2 \}rm https://ethereum.org/smart-contracts/;$ accessed on October 3, 2025

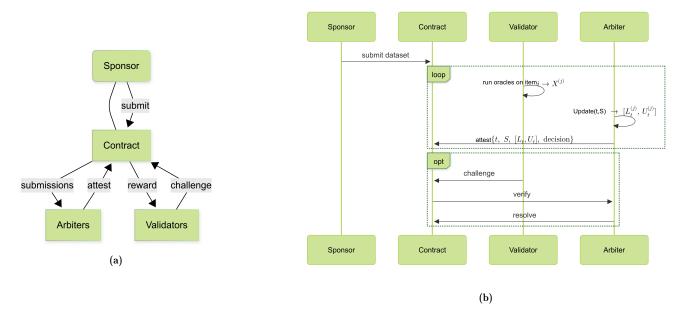


Figure 2: Overview of SIEVE: (a) Global view, (b) Workflow.

- (d) Arbiters reproduce the pack, update t, S_t^j , and call to obtain L_t^j, U_t^j for each P_j (Sec. 3.2).
- (e) Arbiters co-sign attest t, S, L_t, U_t , decision;
- (f) Stopping rule.

$$\text{State} t = \begin{cases} \text{Clean}, & \text{if } \forall j \ : \ U_t^j \leq \varepsilon_j, \\ \\ \text{Dirty}, & \text{if } \exists j \ : \ L_t^j \geq \varepsilon_j, \\ \\ \text{Pending}, & \text{otherwise}. \end{cases}$$

(4) When a terminal decision is reached, the per-property card is stored by content address and referenced on-chain next to rootHash and seed.

SIEVE Confidence Card

dataset : {rootHash, seed}

property: P_j , ε_j , δ_j , oracle_digest

 $\text{evidence}: \quad \left(t,\, S_t^j,\, \widehat{p}_t^j \!\!=\!\! S_t^j t,\, L_t^j, U_t^j\right)$

decision: $\left(\text{State}t, \ \text{T2}\varepsilon_{j} \right)$ if Statet = CLEAN

Reading rule. CLEAN iff $U_t^j \leq \varepsilon_j$; DIRTY iff $L_t^j \geq \varepsilon_j$; otherwise PENDING.

Cleanliness lower bound: $1-U_t^j$ at coverage $1-\delta_j$. Example. Let P_j = buildability, ε_j = 0.5%, $1-\delta_j$ = 95%. Suppose the card shows t = 2,500, S_t^j = $7\Rightarrow \widehat{p}_t^j$ = 0.28%, and L_t^j, U_t^j = 0.13%, 0.48%. Since U_t^j = 0.48% \leq 0.5%, the decision is CLEAN and $T2\varepsilon_j$ = 2,500. The dataset's certified cleanliness for this property is at least $1-U_t^j$ = 99.52% (with 95% anytime coverage).

(5) Validators may challenge arbiters challenge(auditId, t, evidence_uri). The contract records the resolution (and any penalties in incentive-enabled deployments).

By aligning sponsors needs for clear guarantees with an efficient community participation, SIEVE turns ad-hoc, duplicated preprocessing into a transparent, replayable audit. Each dataset version receives a machine-readable *Confidence Card* that (i) states what was checked and with what tolerance, (ii) publishes live, anytime-valid bounds, and (iii) is tamper-resistant (pinned oracles, reproducible sampling, content-addressed records). Thus, we bring less duplicated cleaning (shared, reusable evidence), lower onboarding cost for downstream users (i.e., cards become portable to CI/catalogs), and higher trust for all stakeholders (decisions are auditable and hard to game), without full rescans of the whole dataset.

4 Future Plans

Our future plans focus on operationalizing SIEVE beyond the core statistics (Sec. 3.2) so that (RQ1) evidence is captured with near-zero friction inside developer tools, (RQ2) individual cleaning effort and duplication measurably decrease, and (RQ3) the framework demonstrably delivers value in real-world settings.

4.1 Editor/CI integration (RQ1):

Ship a lightweight SIEVE-Client (VS Code/JetBrains) that opportunistically captures build/test/dependency signals, packages an EvidencePack with one-click consent, and submits it.

4.2 Efficiency & cost (RQ2):

Add cache/skip rules for heavy checks, artifact/layer reuse, and a dashboard that tracks sample efficiency $(T2\varepsilon)$, cleanliness growth $1-U_t$, and cost per certified point.

4.3 Deployment (RQ3):

Run multi-dataset pilots, publish public cards/artefacts (rootHash, seed, oracle, evidences), and wire cards to data catalogs.

Following this plan we expect, reproducible pipeline where editors/CI make evidence "nearly free", cards certify properties with anytime-valid bounds, and pilots show measurable reductions [10] in duplicated cleaning effort and increased trust thus validating the SIEVE for community-driven, per-property dataset certification. [11]

5 Conclusion

We introduced SIEVE, a community-driven framework that turns dataset-quality claims into anytime-valid statistical certificates. without scanning entire datasets. Our goal is to make SIEVE a lightweight yet dependable layer: a card schema, a library, pinned oracles for common properties, and easy editor/CI clients. Dataset hubs and CI systems can consume cards to enforce gates or display cleanliness lower bounds. Practitioners stop rebuilding private filters; instead, they contribute evidence that improves a shared, anytime-verifiable certificate.

6 Acknowledgments

This work is supported by the Luxembourg Ministry of Foreign and European Affairs through their Digital4Development (D4D) portfolio under the project LuxWAyS (Luxembourg/West-Africa Lab for Higher Education Capacity Building in CyberSecurity and Emerging Topics in ICT4Dev.)

References

- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets, 2021. URL https://arxiv.org/abs/1803.09010.
- [2] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. The dataset nutrition label: A framework to drive higher data quality standards, 2018. URL https://arxiv. org/abs/1805.03677.
- [3] Xinyu Yang, Weixin Liang, and James Zou. Navigating dataset documentations in ai: A large-scale analysis of dataset cards on hugging face, 2024. URL https://arxiv.org/abs/2401.13822.
- [4] Anthony Cintron Roman, Jennifer Wortman Vaughan, Valerie See, Steph Ballard, Jehu Torres, Caleb Robinson, and Juan M Lavista Ferres. Open datasheets: Machine-readable documentation for open datasets and responsible ai assessments. arXiv preprint arXiv:2312.06153, 2023.
- [5] Nitisha Jain, Mubashara Akhtar, Joan Giner-Miguelez, Rajat Shinde, Joaquin Vanschoren, Steffen Vogler, Sujata Goswami, Yuhan Rao, Tim Santos, Luis Oala, et al. A standardized machinereadable dataset documentation format for responsible ai. arXiv preprint arXiv:2407.16883, 2024.
- [6] Caleb Geren, Amanda Board, Gaby G Dagher, Tim Andersen, and Jun Zhuang. Blockchain for large language model security and safety: A holistic survey. ACM SIGKDD explorations newsletter, 26(2):1–20, 2025.
- [7] Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union

- legislative acts. https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024.
- [8] Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In Yvette Graham and Matthew Purver, editors, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 67–93, St. Julian's, Malta, March 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-long.5. URL https://aclanthology.org/ 2024.eacl-long.5/.
- [9] Ruchir Puri, David S Kung, Geert Janssen, Wei Zhang, Giacomo Domeniconi, Vladimir Zolotov, Julian Dolby, Jie Chen, Mihir Choudhury, Lindsey Decker, et al. Codenet: A large-scale ai for code dataset for learning a diversity of coding tasks. arXiv preprint arXiv:2105.12655, 2021.
- [10] GitHub and Microsoft Research. Codesearchnet datasets, tools, and benchmarks. https://github.com/github/CodeSearchNet, 2019. Accessed: 2025-09-27.
- OpenAI. Dataset card for openai humaneval. https://huggingface. co/datasets/openai/openai_humaneval, 2021. Accessed: 2025-09-27
- [12] CodeParrot. Apps: Automated programming progress standard dataset card. https://huggingface.co/datasets/codeparrot/apps, 2021. Accessed: 2025-09-27.
- [13] BigCode. The stack v2 dataset card. https://huggingface.co/datasets/bigcode/the-stack-v2, 2024. Accessed: 2025-09-27.
- [14] Siw Elisabeth Hove and Bente Anda. Experiences from conducting semi-structured interviews in empirical software engineering research. In 11th IEEE International Software Metrics Symposium (METRICS'05), pages 10-pp. IEEE, 2005.
- [15] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *Annals of Statistics*, 2021.
- [16] Jean Ville. Etude Critique de la Notion de Collectif. 1939.
- [17] Peter Grünwald, Rianne de Heide, and Wouter Koolen. Safe testing. JRSS B, 2019.