American Journal of Advanced Multidisciplinary Innovation and Research (AJAMIR)

E-ISSN: XXXX-XXXX ● Website: www.ajamir.com ● Email: editor@ajamir.com

Machine-Learning Audit Trails for Reproducible Multidisciplinary Research

Dr. Daniel K. Morgan

Associate Professor, University of Copenhagen, Denmark

Abstract

Machine learning has become integral to multidisciplinary research in health sciences, social science, engineering, environmental studies, computational biology, economics, and other data-intensive fields. Its methodological flexibility creates substantial opportunities for discovery, but it also introduces reproducibility challenges because reported results can depend on dataset versions, preprocessing pipelines, software dependencies, random seeds, hyperparameters, hardware configurations, intermediate transformations, model checkpoints, evaluation scripts, and numerous undocumented analytical decisions. The reproducibility initiative reported by Pineau and colleagues demonstrated the importance of code submission, reproducibility checklists, and structured reporting for strengthening machine-learning research, while FAIR principles extend reproducibility concerns to data, algorithms, tools, and workflows.

This paper proposes a machine-learning audit-trail framework for reproducible multidisciplinary research. An audit trail is conceptualized as a chronological, machine-readable provenance record linking source datasets, preprocessing operations, code versions, software environments, model configurations, training runs, evaluation outputs, human decisions, derived artifacts, and publication claims. The framework integrates principles from W3C PROV, FAIR research practices, dataset documentation, model reporting, experiment tracking, and contemporary provenance systems. A conceptual-methodological research design is accompanied by a transparent simulation involving thirty minimal-audit and thirty comprehensive-audit multidisciplinary workflows. The simulated mean reproducibility score increases from 70.43 under minimal audit trails to 87.53 under comprehensive audit trails, while the median rises from 72.5 to 89.5. These values are illustrative rather than empirical measurements. The analysis suggests that reproducibility is strengthened when provenance capture extends beyond source-code availability to include datasets, execution environments, experiment lineage, parameter choices, unsuccessful runs, evaluation procedures, and links between computational artifacts and manuscript claims. The study concludes that audit trails should become persistent research infrastructure rather than retrospective documentation assembled only at publication.

Keywords: machine learning, audit trails, reproducibility, research provenance, multidisciplinary research, experiment tracking, data lineage, model versioning, FAIR principles, research integrity

American Journal of Advanced Multidisciplinary Innovation and Research (AJAMIR)

E-ISSN: XXXX-XXXX ● Website: www.ajamir.com ● Email: editor@ajamir.com

1. Introduction

Machine learning increasingly operates as a methodological bridge among disciplines that historically relied on different forms of evidence, analytical software, and research cultures. Clinical researchers may combine imaging and electronic health records, environmental scientists may integrate remote-sensing and sensor streams, engineers may construct predictive models from experimental measurements, and social scientists may combine surveys with administrative or digital behavioral data. This interdisciplinarity expands analytical possibilities but also increases the complexity of reproducing computational findings. Reproducibility in computational research generally requires that consistent results can be obtained using the same data, computational methods, code, and analytical conditions. The National Academies' treatment of computational reproducibility similarly emphasizes consistent results using the same input data, computational steps, methods, code, and conditions of analysis.

Machine-learning research adds additional complexity because an apparently simple statement that a model was “trained on dataset X” conceals a substantial chain of technical decisions. Dataset X may have multiple releases; records may have been filtered or imputed; categorical variables may have been encoded differently across experiments; training and evaluation splits may have been generated randomly; software-library versions may alter numerical behavior; and model performance may depend on random initialization, hardware, hyperparameter searches, or selection of a particular checkpoint. Pineau and colleagues' report on the NeurIPS 2019 Reproducibility Program formalized several of these concerns through code-submission practices, a reproducibility challenge, and a machine-learning reproducibility checklist. The broader lesson is that a published model result is the endpoint of a computational process whose intermediate states often determine whether another researcher can reproduce the result.

These problems become more serious in multidisciplinary research because collaborators may interpret the same artifacts differently. A domain researcher may consider the original dataset the authoritative research object, whereas a machine-learning scientist may work primarily from a processed tensor or feature table generated several steps downstream. A statistician may require information about sampling and exclusion rules, while a software engineer may require dependency specifications, container configurations, and repository commits. FAIR principles provide a useful foundation because they explicitly extend Findability, Accessibility, Interoperability, and Reusability beyond conventional data to algorithms, tools, and workflows. FAIR principles for research software similarly recognize that reusable computational research depends upon treating software itself as a managed scholarly object.

Provenance offers a technical and conceptual mechanism for connecting these heterogeneous components. The W3C PROV Data Model defines provenance through relationships among entities, activities, and agents and is intended to support assessments concerning quality, reliability, and trustworthiness. PROV-O provides an interoperable ontology for representing and exchanging such information across systems and domains. More recent machine-learning provenance research has extended similar concepts toward end-to-end tracking of datasets, experiments, workflows, models, and governance records. Padovani and colleagues' yProv4ML, for example, was designed to collect machine-learning provenance in formats compatible with W3C PROV and ProvML. Contemporary research on audit trails further argues that lifecycle events should be connected in chronological,

durable, and reviewable records capable of showing what changed, when it changed, and who or what was responsible.

Against this background, the present paper develops a machine-learning audit-trail architecture specifically for reproducible multidisciplinary scholarship. The approach differs from conventional experiment tracking by connecting technical lineage with the evidentiary structure of the research manuscript. The proposed trail extends from source-data acquisition through preprocessing, model development, evaluation, and artifact generation to the exact tables, graphs, and claims presented in a publication. The study additionally introduces a simulation-based comparison between minimal and comprehensive audit-trail practices. No actual multidisciplinary research teams were observed, and the numerical analysis is explicitly synthetic. The objective is methodological: to demonstrate how provenance completeness can be operationalized and how audit infrastructure can improve the capacity of researchers, reviewers, collaborators, and institutions to reconstruct machine-learning research.

2. Objectives of the Study

2.1 To Define a Complete Machine-Learning Audit Trail

The first objective is to identify the minimum chain of evidence required to reconstruct a machine-learning experiment from its source data to its published research claim. The proposed audit trail includes dataset identifiers, acquisition dates, preprocessing functions, transformation parameters, code commits, dependency environments, hardware information where relevant, random seeds, hyperparameters, model checkpoints, metrics, evaluation datasets, statistical analysis, and generated outputs.

2.2 To Evaluate Audit-Trail Requirements Across Disciplines

The second objective is to establish an audit structure sufficiently general to support multidisciplinary research without ignoring discipline-specific requirements. The framework therefore separates universal machine-learning provenance from domain extensions.

2.3 To Develop a Reproducibility-Oriented Governance Model

The third objective is to connect audit trails with research integrity and institutional governance. An effective audit trail should support internal verification before submission, independent reproduction after publication, investigation of discrepancies, correction of computational errors, and determination of which downstream claims were affected when a dataset, model, or analytical artifact changes.

.

3. Review of Literature

3.1 Reproducibility Challenges in Machine-Learning Research

Reproducibility has become a significant methodological concern within machine learning because reported performance can depend on implementation details that are inadequately described in conventional manuscripts. Pineau and colleagues' NeurIPS reproducibility initiative represents one of the field's most influential systematic responses. The initiative combined a code-submission policy, reproducibility challenge, and reproducibility checklist intended to improve the conduct and reporting of machine-learning research. The significance of this work extends beyond conference administration

because it recognizes that reproducibility requires explicit disclosure of experimental conditions rather than relying on readers to infer them.

The problem is particularly visible in stochastic machine-learning methods. Henderson and colleagues demonstrated substantial variability in deep reinforcement-learning results arising from implementation choices, stochasticity, hyperparameters, and experimental practices, arguing for stronger statistical reporting and standardized evaluation. Similar concerns appear in applied fields. McDermott and colleagues reviewed more than one hundred machine-learning-for-health papers and identified weaknesses in reproducibility-related practices, particularly concerning code and data accessibility. These examples show that reproducibility difficulties can propagate from core machine learning into disciplines adopting machine learning as an analytical instrument.

Semmelrock and colleagues' review of machine-learning-driven research identifies unpublished code or data and sensitivity to training conditions among continuing barriers to reproducibility. This is important for multidisciplinary scholarship because researchers outside computer science may use machine learning without having established conventions for recording model-development decisions. A manuscript may report an algorithm and several hyperparameters while omitting the precise preprocessing sequence, package versions, model-selection history, or intermediate dataset construction required to regenerate the final analysis.

3.2 Provenance, FAIR Research Objects, and Machine-Learning Lineage

Research provenance addresses this chain by recording the origins and transformations of computational artifacts. W3C PROV provides an interoperable conceptual model based principally on entities, activities, and agents. A dataset can be represented as an entity, preprocessing as an activity, a researcher or software service as an agent, and the generated feature table as a derived entity. The same structure can continue through model training, evaluation, and publication.

Samuel, Löffler, and König-Ries explicitly connected machine-learning pipelines with provenance, reproducibility, and FAIR data practices, emphasizing that source code and datasets alone do not capture all factors necessary for reproducing ML experiments. Souza and colleagues likewise proposed workflow provenance across the lifecycle of scientific machine learning, emphasizing end-to-end relationships between raw scientific data, processing workflows, and learned models. More recent provenance systems such as yProv4ML extend this approach through machine-readable capture designed to integrate machine-learning runs with wider scientific workflows.

FAIR principles strengthen the same logic from a research-management perspective. Wilkinson and colleagues intended FAIR principles to apply not only to conventional datasets but also to algorithms, tools, and workflows. FAIR4RS subsequently adapted FAIR thinking specifically to research software. In multidisciplinary research, this implies that reproducibility infrastructure should manage software versions and computational artifacts with the same seriousness traditionally applied to primary research data.

3.3 Experiment Tracking, Auditability, and Multidisciplinary Collaboration

Modern experiment-tracking tools operationalize several components of provenance. MLflow Tracking, for example, supports logging parameters, metrics, runs, and artifacts, while MLflow's dataset-tracking

functionality explicitly supports dataset versioning and lineage intended to improve experiment reproducibility as underlying data evolve. Model-registry functionality can additionally maintain links among models, experiments, datasets, and development artifacts. Such platforms demonstrate that provenance can be captured automatically rather than relying exclusively on researchers to create retrospective documentation.

However, experiment tracking and scholarly reproducibility are not identical. An experiment tracker may contain hundreds of runs but provide no direct link between those runs and the performance value eventually quoted in a manuscript. Failed experiments may be deleted or overlooked. A researcher may export a graph and modify it in separate software, breaking the chain between source metrics and the published image. A model checkpoint may remain available without the version of the preprocessing code that produced its input. Consequently, an audit trail needs an additional claim-to-artifact layer.

Benchopt provides an example of reproducibility-centered collaborative infrastructure by automating and publishing optimization benchmarks across programming languages. Recent Rollout Cards research similarly argues that preserving underlying evaluation records and reporting rules is necessary because apparently minor reporting choices can materially change reported benchmark results. Evaluation Cards extend this direction by combining benchmark metadata, evaluation runs, model metadata, documentation completeness, provenance, and comparability into a unified reporting structure.

The emerging lesson is that reproducible multidisciplinary research requires a record of both what computationally occurred and how those computational events became scientific evidence. This second relationship is especially important when multiple disciplines collaborate because different team members may operate at different points in the pipeline. A robust audit trail can provide a common evidentiary language linking domain data, analytical code, model outputs, statistical interpretation, and publication.

4. Research Methodology

4.1 Research Design

The study adopts a conceptual-methodological research design with simulation-based reproducibility analysis. No actual laboratory groups, research institutions, confidential datasets, or published projects were audited. The numerical results reported in the Analysis section were constructed exclusively for methodological demonstration.

The conceptual framework was developed through synthesis of primary literature and authoritative standards concerning machine-learning reproducibility, provenance, FAIR research practices, model and dataset documentation, experiment tracking, and AI lifecycle governance. Particular attention was given to research proposing reproducibility checklists, workflow provenance models, dataset and model documentation standards, W3C provenance representations, and automated experiment-lineage systems.

American Journal of Advanced Multidisciplinary Innovation and Research (AJAMIR)

E-ISSN: XXXX-XXXX ● Website: www.ajamir.com ● Email: editor@ajamir.com

4.2 Machine-Learning Audit-Trail Architecture

The proposed audit architecture organizes provenance into eight connected layers.

Table 1: Proposed Machine-Learning Audit-Trail Framework

Audit Layer	Required Audit Evidence	Example Recorded Elements	Reproducibility Function
Source Data	Persistent identity and provenance of research inputs	Dataset ID, version, source, acquisition date, license, checksum	Ensures researchers use the same source evidence
Data Transformation	Complete record of preprocessing and feature construction	Filtering, imputation, normalization, augmentation, exclusions	Reconstructs the analytical dataset
Code Provenance	Versioned computational implementation	Repository, commit hash, branch, uncommitted changes	Identifies the exact analytical code
Execution Environment	Software and computational configuration	OS, Python/R version, packages, containers, accelerators	Reduces environment-dependent discrepancies
Experiment Configuration	Parameters governing model training	Hyperparameters, random seeds, splits, configuration files	Recreates training conditions
Run and Model Lineage	Relationship between runs and trained artifacts	Run ID, checkpoint, parent experiment, model hash	Identifies which experiment created which model

Source: Author-developed framework informed by W3C PROV, FAIR principles, reproducibility research, experiment tracking, Datasheets for Datasets, Model Cards, and recent ML audit-trail scholarship.

The fundamental provenance structure can be represented as:

[D_v \rightarrow P_c \rightarrow C_h \rightarrow E_s \rightarrow R_i \rightarrow M_j \rightarrow V_k \rightarrow A_l \rightarrow C_m]

where:

(D_v) = versioned source dataset (P_c) = preprocessing configuration (C_h) = code commit or immutable code identifier (E_s) = execution environment specification (R_i) = experiment run (M_j) = trained model artifact (V_k) = validation or evaluation result (A_l) = publication artifact (C_m) = manuscript claim

This structure turns reproducibility into a traversable graph rather than a collection of disconnected files.

American Journal of Advanced Multidisciplinary Innovation and Research (AJAMIR)

E-ISSN: XXXX-XXXX ● Website: www.ajamir.com ● Email: editor@ajamir.com

4.3 Simulation and Reproducibility Assessment

Two illustrative conditions were created, each representing thirty multidisciplinary workflows.

The Minimal Audit Trail records basic source-code access, final training parameters, and headline performance but inconsistently preserves dataset versions, preprocessing lineage, random seeds, execution environments, failed runs, or claim-level links.

The Comprehensive Audit Trail records versioned inputs, preprocessing operations, code commits, dependency environments, random seeds, experiment runs, models, evaluation scripts, and manuscript-artifact lineage.

A conceptual Reproducibility Audit Score (RAS) is proposed:

[RAS = 0.20D + 0.15C + 0.15E + 0.15X + 0.15M + 0.10V + 0.10P]

where:

(D) = data lineage completeness (C) = code-version completeness (E) = execution-environment completeness (X) = experiment-configuration completeness (M) = model and run lineage (V) = evaluation traceability (P) = publication-claim provenance

All dimensions are normalized from 0 to 100. The weighting is an illustrative methodological proposal and has not been externally validated.

5. Analysis

5.1 Comparison of Minimal and Comprehensive Audit Trails

The simulated analysis shows a clear difference between the two audit-trail conditions.

Table 2: Simulated Reproducibility Outcomes by Audit-Trail Maturity

Audit-Trail Model	Number of Simulated Workflows	Mean Reproducibility Score	Median	Minimum	Maximum	Primary Weakness
Minimal Audit Trail	30	70.43	72.5	42	88	Missing lineage between datasets, preprocessing, environment and reported findings
Comprehensive Audit Trail	30	87.53	89.5	68	98	Residual dependence on unavailable external resources or nondeterministic components

American Journal of Advanced Multidisciplinary Innovation and Research (AJAMIR)

E-ISSN: XXXX-XXXX ● Website: www.ajamir.com ● Email: editor@ajamir.com

Note: These data are simulated. They do not report audits of actual researchers or institutions.

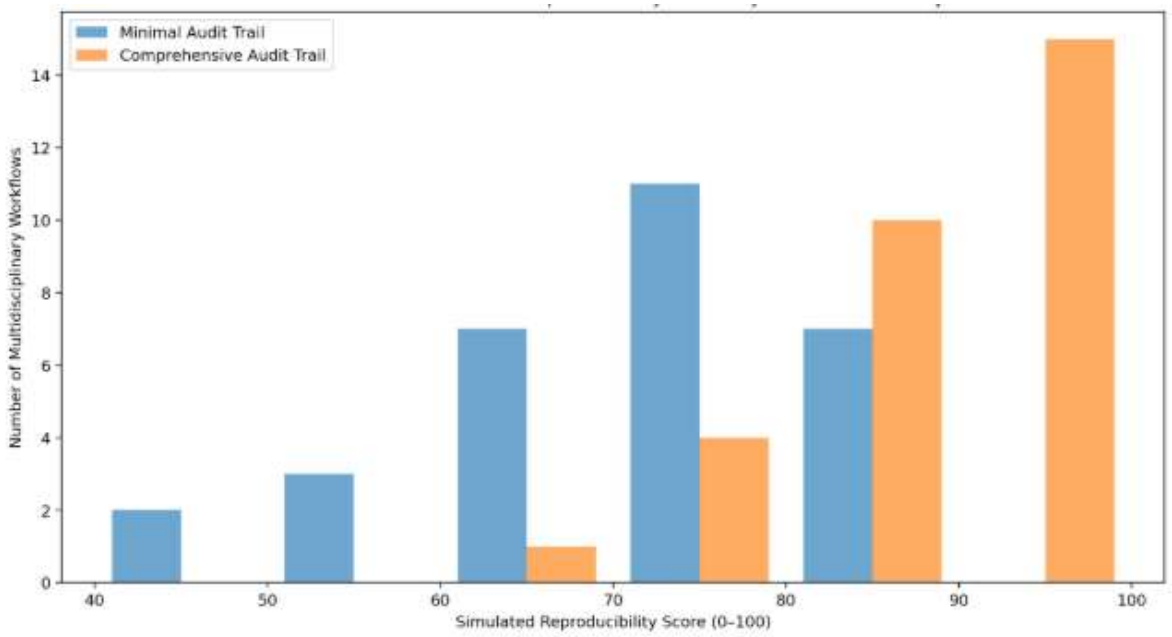
The comprehensive condition records a mean advantage of approximately 17.1 points over the minimal condition. The median difference is also substantial, increasing from 72.5 to 89.5. The weakest minimal-audit workflow scores only 42, illustrating how apparently documented projects can remain difficult to reproduce when key intermediate states are missing.

The difference does not arise from source-code availability alone. In the conceptual framework, both conditions may preserve code. The stronger condition performs better because it preserves the identity and relationship of artifacts.

For example, access to a repository is insufficient when the reader cannot determine which commit generated the final model. A dataset filename is insufficient when the underlying dataset changed over time. A reported hyperparameter value is insufficient when the train-test split or random seed is unknown. A saved model is insufficient when the preprocessing transformation that produced its features has disappeared.

5.2 Histogram of Simulated Reproducibility Scores

Graph 1: Distribution of Simulated Reproducibility Scores by Machine-Learning Audit-Trail Maturity



Source: Author-generated simulation.

The histogram reveals an important difference in the shape of the simulated distributions. Minimal-audit workflows are spread widely across moderate and higher reproducibility ranges, indicating substantial variation in whether an independent researcher could reconstruct the computational procedure. Comprehensive-audit workflows shift markedly toward the upper end of the scale, with most cases falling between approximately 80 and 100.

This simulated result illustrates why reproducibility infrastructure should aim not only to increase average documentation quality but also to reduce variation between projects. In multidisciplinary environments, inconsistent documentation standards create substantial hidden costs. One project may maintain detailed environment files while another records only package names. One laboratory may version datasets while another silently overwrites them. One researcher may preserve every experiment run while another keeps only the best model.

5.3 Claim-Level Reproducibility and Audit Reconstruction

The final analytical issue concerns the relationship between computational artifacts and published claims. A conventional audit may establish that a model exists and that the training process can be rerun.

Scientific reproducibility requires an additional question:

Can the exact evidence supporting a published claim be reconstructed?

Consider a manuscript reporting:

"Model B improved F1 score by 4.2 percentage points over Model A."

A complete audit trail should permit a reviewer to retrieve the dataset version, evaluation cohort, Model A artifact, Model B artifact, metric implementation, random-seed set, statistical aggregation rule, resulting metric file, manuscript table cell, and any graphical representation derived from the comparison.

This form of claim-level provenance is particularly important when an upstream error is discovered. If a preprocessing defect affects dataset version 3.4, the audit graph can identify every downstream model trained on that version and every manuscript result derived from those models.

Recent audit-trail research emphasizes a similar capacity for reconstruction by connecting lifecycle events, technical provenance, governance actions, and change history. W3C PROV provides a suitable conceptual foundation because provenance relationships can connect entities and activities across heterogeneous systems.

6. Discussion

6.1 Reproducibility Requires Connected Provenance Rather Than More Documentation

The central implication of the study is that reproducibility failures should not be understood solely as failures to document enough information. Research teams frequently generate substantial documentation: laboratory notes, Git repositories, data dictionaries, configuration files, experiment dashboards, cloud logs, notebooks, statistical outputs, and manuscript drafts. The deeper problem is that these artifacts may not be connected sufficiently to reveal which specific combination generated the published result. Provenance converts documentation into an evidentiary network by recording derivation and dependency relationships. W3C PROV is useful precisely because it represents entities, activities, and responsible agents as connected components rather than isolated metadata fields.

This distinction explains why experiment-tracking systems are valuable but not sufficient in isolation. MLflow can record parameters, metrics, datasets, runs, artifacts, and model lineage, providing important technical infrastructure for reproducibility. Yet the scholarly workflow can continue outside the tracking system. Results may be copied into spreadsheets, reformatted, statistically combined, or inserted manually into manuscripts. A complete audit architecture should therefore assign persistent

identities to important derived artifacts and preserve the relationship between computational evidence and scholarly communication.

Datasheets and Model Cards serve a complementary function because some essential reproducibility information cannot be understood from execution logs alone. A dataset may be technically reproducible while being inappropriate for the research question because important inclusion criteria or limitations were not documented. Datasheets address the history, composition, collection, and intended use of datasets, while Model Cards provide context about intended model use and evaluation conditions. A mature audit trail should link these descriptive documents to machine-readable lineage rather than choosing between them.

The broader principle is that reproducibility requires three forms of evidence simultaneously: descriptive evidence explaining the research objects, operational evidence recording what happened, and relational evidence showing how one artifact produced another. Removing any one of these layers weakens the ability of another researcher to reconstruct the study.

6.2 Multidisciplinary Research Requires a Common Provenance Core with Domain Extensions

Multidisciplinary collaboration makes standardization necessary but also makes excessively rigid standards difficult. A universal machine-learning audit trail cannot anticipate every relevant variable in medicine, environmental science, materials engineering, economics, psychology, remote sensing, or computational biology. The solution is therefore not to construct a single enormous checklist for every discipline. A more sustainable strategy is to create a stable provenance core that can be extended by domain-specific modules.

The core should record dataset identity, transformations, code, execution environment, experiment configuration, model artifacts, evaluation, and publication lineage. These components are relevant regardless of whether the model predicts cardiovascular outcomes, crop yields, structural fatigue, financial risk, or ecological change. Domain extensions can then provide specialized metadata. Medical research may add cohort selection and imaging-device parameters. Remote-sensing research may preserve spatial resolution and coordinate-reference systems. Materials research may preserve specimen composition and experimental conditions. Social research may record weighting, missingness rules, and survey instruments.

FAIR principles provide strong conceptual support for such interoperability because they seek to make digital research objects findable, accessible, interoperable, and reusable, while FAIR4RS specifically extends this logic to software. W3C PROV similarly allows domain-specific extensions while preserving a common provenance representation.

6.3 Audit Trails Should Capture Decisions, Failures, and Research Evolution

Another important implication concerns the common practice of preserving successful experiments while discarding unsuccessful ones. Machine-learning research frequently involves exploratory branching: numerous model configurations are trained, preprocessing alternatives are tested, different metrics are considered, and several datasets may be evaluated before the reported workflow is selected. If only the final configuration survives, reviewers cannot determine the breadth of model selection or whether the reported result emerged from extensive search.

Recent reproducibility proposals in agent research demonstrate the importance of preserving underlying runs, failures, skipped cases, and reporting rules rather than storing only final aggregate scores. Evaluation Cards similarly emphasize that benchmark and evaluation reporting requires provenance and documentation sufficient to interpret aggregate claims. Although these frameworks target particular AI settings, their underlying principle applies widely: the audit trail should preserve scientifically meaningful information about how the reported result was selected.

This does not mean every temporary debugging operation must become a permanent scholarly artifact. Such an approach would create excessive storage and governance burdens. Instead, audit policies should distinguish operational telemetry from research-significant events. A change to a model architecture, dataset version, preprocessing rule, evaluation population, random-seed policy, or primary metric should ordinarily be preserved because it can affect scientific interpretation.

7. Conclusion

Machine learning has expanded the analytical capacity of multidisciplinary science, but the same flexibility that enables complex discovery also creates reproducibility challenges. A modern machine-learning result is rarely produced by a single script operating on a single fixed dataset. It emerges from interacting data versions, transformations, software environments, model configurations, stochastic runs, evaluation procedures, and human analytical choices. Releasing source code and a final model therefore represents only part of the evidence required to reconstruct the research. The reproducibility initiatives developed within the machine-learning community, together with FAIR research practices and provenance standards, show that reliable computational scholarship increasingly depends upon systematic preservation of the research process itself.

The framework proposed in this study addresses this requirement by treating an audit trail as a chronological and relational record extending from source data to published claim. Its eight layers include source-data identity, transformation provenance, code versioning, computational environments, experiment configurations, run and model lineage, evaluation evidence, and publication-claim linkage. The simulation demonstrates the methodological value of this architecture: workflows operating under comprehensive audit-trail assumptions produced substantially higher simulated reproducibility scores than workflows preserving only minimal evidence. The mean score increased from 70.43 to 87.53, while the simulated median increased from 72.5 to 89.5. These values are not empirical measurements, but they illustrate how reproducibility can improve when missing provenance links are systematically addressed.

The most important contribution is the introduction of claim-level lineage. Reproducing a machine-learning model is necessary but does not automatically reproduce the scientific argument built from that model. Tables, graphs, statistical summaries, and manuscript statements should therefore remain linked to the computational evidence from which they were generated. Such lineage becomes especially valuable when errors are identified after publication because researchers can trace affected downstream artifacts rather than reassessing an entire project manually. Provenance standards, experiment-tracking systems, dataset documentation, model reporting, and research-software FAIR practices already provide many of the technical building blocks required for this approach.

Machine-learning audit trails should consequently be understood not as additional bureaucracy but as research infrastructure supporting scientific memory. Future empirical work should evaluate audit-trail adoption across multidisciplinary laboratories, measure the time required for independent reproduction, compare automated and manual provenance capture, and assess storage, privacy, intellectual-property, and security trade-offs. Research should also investigate cryptographic integrity mechanisms, standardized provenance interchange, and automated generation of publication-level reproducibility packages. The long-term objective should be a research ecosystem in which a published machine-learning result can be followed backward through every significant computational dependency and forward into every derived scientific claim. Such an infrastructure would strengthen reproducibility, research integrity, collaboration, peer review, and confidence in machine-learning-assisted scholarship.

References

1. Peng, R. D. (2011). Reproducible research in computational science. *Science, 334*(6060), 1226–1227. https://doi.org/10.1126/science.1213847
2. Stodden, V., Leisch, F., & Peng, R. D. (Eds.). (2014). *Implementing Reproducible Research*. Chapman and Hall/CRC.
3. Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology, 9*(10), e1003285. https://doi.org/10.1371/journal.pcbi.1003285
4. Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., & Teal, T. K. (2017). Good enough practices in scientific computing. *PLoS Computational Biology, 13*(6), e1005510. https://doi.org/10.1371/journal.pcbi.1005510
5. Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review, 49*(1), 71–79. https://doi.org/10.1145/2723872.2723882
6. Stodden, V., & Miguez, S. (2014). Best practices for computational science: Software infrastructure and environments for reproducible and executable research. *Journal of Open Research Software, 2*(1), e21. https://doi.org/10.5334/jors.ay
7. Piccolo, S. R., & Frampton, M. B. (2016). Tools and techniques for computational reproducibility. *GigaScience, 5*, 30. https://doi.org/10.1186/s13742-016-0159-4
8. Rule, A., Birmingham, A., Zuniga, C., Altintas, I., Huang, S.-C., Knight, R., Moshiri, N., Nguyen, M. H., Rosenthal, S. B., & Rose, P. W. (2019). Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLoS Computational Biology, 15*(7), e1007007. https://doi.org/10.1371/journal.pcbi.1007007
9. Simmhan, Y. L., Plale, B., & Gannon, D. (2005). A survey of data provenance in e-science. *ACM SIGMOD Record, 34*(3), 31–36. https://doi.org/10.1145/1084805.1084812
10. Moreau, L., & Missier, P. (Eds.). (2013). *PROV-DM: The PROV Data Model*. W3C Recommendation.
11. Moreau, L., Freire, J., Futrelle, J., McGrath, R. E., Myers, J., & Paulson, P. (2008). The Open Provenance Model: An overview. *Lecture Notes in Computer Science, 5272*, 323–326. https://doi.org/10.1007/978-3-540-89965-5_31

12. Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data, 3*, 160018. https://doi.org/10.1038/sdata.2016.18
13. Bietz, M. J., Coppersmith, D., & Birnbaum, M. (2016). Utility, sovereignty, and individual data ownership in scientific research. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 1–12.
14. Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., Fox, E., & Larochelle, H. (2021). Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research, 22*(164), 1–20.
15. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems, 28*, 2503–2511.
16. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Gebru, T., & Kleinberg, J. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229. https://doi.org/10.1145/3287560.3287596
17. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM, 64*(12), 86–92. https://doi.org/10.1145/3458723
18. Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Drozdzal, M., & Roy, A. (2019). Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems, 1*, 747–758.
19. Henderson, P., Soyer, H., Strub, F., Antonoglou, I., Bates, R., & Pineau, J. (2018). Deep reinforcement learning that matters. *Proceedings of the AAAI Conference on Artificial Intelligence, 32*(1), 3207–3214.
20. Kapoor, S., & Narayanan, A. (2021). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns, 2*(9), 100434. https://doi.org/10.1016/j.patter.2021.100434