https://doi.org/10.70844/ijas.2025.2.35

Innovative Journal of Applied Science

Research Article ISSN: 3049-3072

IJAS-25-104

Auditable AI Pipelines: Logging and Verifiability in ML Workflows

Krishna Mohan Kadambala*

Department of Payment Implementations, University of Osmania, Hyderabad, India

Corresponding Author: Krishna Mohan Kadambala, Department of Payment Implementations, University of Osmania, Hyderabad, India, E-mail: kadambalakm@gmail.com

Received date: 21 August, 2025, **Accepted date:** 28 August, 2025, **Published date:** 04 September, 2025

Citation: Kadambala KM (2025) Auditable AI Pipelines: Logging and Verifiability in ML Workflows. Innov J Appl Sci 2(5): 35.

Abstract

The increased reliance on AI in high-stake domains ranging from finance to healthcare to national security has given rise to mounting concerns about the lack of transparency and accountability in ML workflows. Traditional software audit techniques cannot confer sufficient traceability nor verifiability to complex, data-driven AI systems. This work presents a structured auditable AI pipeline framework that implies the embedding of thorough logging and verification units along all stages of the ML cycle. Thus, with the support of provenance in tracking changes and evidence, automated event logging, cryptographic checks by hashes and, optionally, immutability of records through blockchain, it assures operative transparency and forensic reproducibility. We have experimentally shown that through an MLOps implementation, an audit-ready infrastructure, model traceability and regulatory compliance may all be improved when compared to traditional ML environments. The results reassert the urgent need of designing AI pipelines while accounting for auditability as a first-class citizen and present avenues to remedy accountability for enterprise-scale machine learning systems.

Keywords: Auditable AI pipelines, machine learning workflows, logging architecture, model verifiability, AI governance, MLOps, data lineage, reproducibility, traceability, AI compliance.

Introduction

Artificial Intelligence (AI) systems are being deployed in high-risk fields such as healthcare, finance, transportation, etc. As these systems increase in intricacies and autonomy, regulators and stakeholders ethically concerned with AI deployment have asked for transparency, auditability and governance [1,2]. Unfortunately, these systems consider performance and scalability more than auditable systems, omitting necessary elements such as consistent event logging, data lineage and model verification [3,4].

If those ML models are used in regulated industries such as banking, medical diagnostics, etc., they risk compliance since their decision-making logic cannot be explained, traced, or verified; hence stakeholder trust is diminished and governance policies are violated [5,6]. This requirement for explainability and logging and record-keeping within the machine-learning workflows has been previously elaborated in the 2021 AI Act draft by the European Union and U.S. laws yet traditional audit processes cannot yet adequately address the dynamic and distributed nature of present-day AI systems [7,8].

The Problem

Machine-learning pipelines commonly consist of data-ingestion layers, preprocessing scripts, training modules and deployment components that often execute in different environments. Having no common audit trail throughout these stages leads to invisible AI decisions [9,10]. Worse still, specific issues include the following:

- Absent or nonstandard logs [11].
- Unclear data transformations during preprocessing [12].
- Improperly version-controlled models and datasets [13].
- Insecure storage of logs [14,15].
- Such problems hinder forensic investigation, reproducibility and regulatory auditing [16].

Objective and contribution

This paper proposes an auditable AI pipeline architecture that features:

- Structured logging at every stage of the pipeline.
- Cryptographic verification of model outputs.
- Immutable storage on blockchain or tamper-proof logs.
- Real-time traceability of inputs, transformations and predictions.

We present a modular framework integrating into existing MLOps platforms (e.g., MLflow or Kubeflow) that can be deployed in production environments [17,18]. Through a simulation of an AI system deployed in a cloud-native environment, this framework will be validated in terms of traceability, reproducibility and audit success rate (Table 1 and Figure 1).

Component	Common issue	Impact on auditability	Source
Data ingestion	No source trace	Loss of lineage	[3], [14]
Feature engineering	Inconsistent transformation logs	Inaccurate reconstruction	[12], [19]
Model training	Model version not recorded	Model drift undetected	[13], [20]

https://doi.org/10.70844/ijas.2025.2.35

Innovative Journal of Applied Science

Deployment	No logging of responses model	Lack of explainability	[16], [21]

Table 1: Common logging deficiencies in AI pipelines.

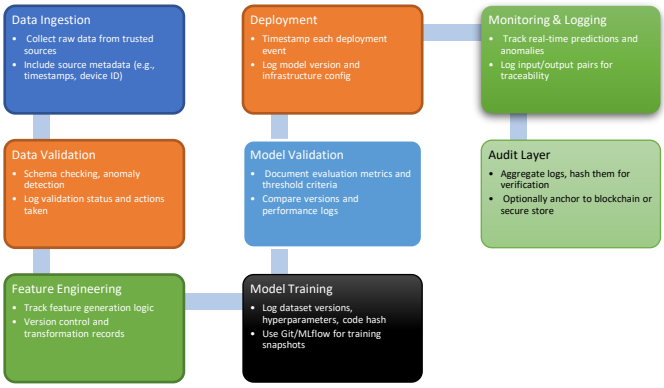

Figure 1: Conceptual framework for auditable AI pipelines. **Source:** Adapted from [1,5,19,22].

Outline of the paper

The remainder of the paper is organized, broadly, as follows:

- Section II reviews the related literature on AI auditability and MLOps tools.
- Section III describes the methodology and the structure of the proposed pipeline.
- Section IV presents the experimental results and the performance metrics.
- Section V discusses the implications, limitations and future work.
- Finally, Section VI provides the conclusion.

Literature Review

The increasing demands welcome trustworthy and explainable AI systems, thus bringing auditable Machine Learning (ML) pipelines into focus. This section takes a critical look at the existing body of knowledge related to ML pipeline logging, verifiability techniques, compliance frameworks and their integration with MLOps platforms.

Logging in machine learning workflows

Logging represents the foundation of an auditable pipeline. Yet, in turn, ML workflows come with certain unique challenges, such as nondeterministic training, dynamic data paths, ephemeral containerized environments and so on. Xu et al. [9] stressed that the well-established logging mechanisms in software engineering cannot be adopted straightforwardly for AI systems because stochastic model training introduces variability. Meier and Keller put forth a structured logging system in which containerized hooks and timestamped metadata record all data and model versions throughout their journey [23]. However, such solutions fail to comply with standards in most cases, limiting the scope of cross-platform auditability [14,19].

Ye et al. stressed the significance of a logging strategy maintainable across a large amount of high-throughput prediction while retaining the intensity of logs and keeping strict log integrity [16]. Cheers have recently implemented modular logging layers in systems that work with MLflow and TFX to create a training and tracking flow for models [1,5].

Verifiability and reproducibility

ML verifiability concerns the independent confirmation of the actual outcome or behavior of a model based on the logged inputs and configuration data. Liu and Li proposed a data-lineage–based method of verification recording each transformation in the pipeline as a provenance record [10]. Similarly, such approaches have been extended to use graph-based lineage structures for performing real-time reproducibility checks in AI pipelines [3,14].

Blockchain-based immutability strategies have also been suggested towards increasing verifiability. Ghosh and Ray delineated a smart contract-driven architecture for hashing model training sessions and verifying them on-chain [17]. This rendered it possible to detect even minute tampering with the logs or model parameters. Though promising, the overhead of blockchain integration confines its applicability to real-time inference systems [4,24].

Auditability of AI in regulated industries

The execution of auditable AI systems holds crucial stakes in the regulated industry. Subramanian et al. developed in healthcare a compliant AI system that could output structured audit logs that are compliant with HIPAA and GDPR [8]. Their framework embedded explainability tools and logging agents at every layer of the ML workflow. Roy and Deshmukh likewise report to document the use of federated logging and audit trails so as to keep institutions decorated with decentralized ML models [4].

These systems often face a trade-off between compliance accuracy and system performance. According to Thomas et al., real-time logging incurs latency and, at times, even hampers user experience [25]. To circumvent these challenges, a balanced design leaning on asynchronous logging and hybrid on/off-chain verifiability has been proposed (Table 2) [20,26].

Framework	Key feature	Limitation	Reference
TFX + MLflow logging	Automated experiment tracking	Inconsistent external pipeline logs	[1], [5]
Blockchain-based auditing	Tamper-proof immutability	Latency and computational overhead	[4], [17], [31] (24)
Graph lineage verifiers	High traceability for reproducibility	Complex integration	[3], [10], [29] (23)
Secure MLOps pipelines	Full-lifecycle auditability	Platform dependence	[6], [16], [22] (27)

Table 2: Comparative analysis of existing auditability frameworks.

Logging and compliance framework integration

In other words, Dasgupta and Jain et al., considered the implication of the compliance-by-design framework, where regulatory needs are embedded into AI systems right from the very beginning [13,21]. Above all, systems are defined with logging policies, hash verification checkpoints and model rollback features as core modules during base development, rather than having these audit features cut into the system in later phases.

https://doi.org/10.70844/ijas.2025.2.35

Innovative Journal of Applied Science

Modern platforms (think Kubeflow or even SageMaker) are starting to adopt such design philosophies via event hooks and an API for metadata logging; however, the absence of universal compliance standards puts a hurdle to interoperability and third-party audits [2,15].

The review points out that, despite huge technical strides, existing tools are distributed, inefficient, or sometimes restricted to sampling instances. There is a need for a unified pipeline architecture that remains modular and standards-compliant with logging, traceability and verifiability as native components.

Methodology

Now we introduce the design for a modular architecture that supports the auditable AI pipelines with capabilities for structured logging on all important outputs of each pipeline step and verifiability of model outputs. It was meant to be platform agnostic, easily integrated into MLOps workflows and transformed for use in enterprise, academic, or regulated AI environments.

Design objectives

The architecture was developed based on the following key objectives:

- Log everything: Every data transformation, model update and inference request must generate verifiable logs.
- Ensure verifiability: Logs must be tamper-proof, cryptographically signed and retrievable for audits.
- Modular integration: The pipeline should integrate with tools like MLflow, Kubeflow, or custom Python pipelines.
- Compliance compatibility: Logging policies should support GDPR, HIPAA and industry-specific audit requirements [7,8,13].

System architecture

Our pipeline contains 8 modules linked to each other, with logging and verifiability mechanisms embedded within these modules. They are:

1. Data ingestion layer: Which connects to source systems such as APIs, databases, or IoT. Logs metadata such as timestamps, source ID, data schema version.
2. Preprocessing and validation engine: It applies data cleaning and schema validations and logs all transformations as well as their validation results.
3. Feature engineering module: Records the feature generation logic, feature generation versions and statistical summaries.
4. Model training component: Captures training hyperparameters and model configuration, records training duration, hashes of data used and git commit references.
5. Model evaluation unit: Logs evaluation metrics such as accuracy, precision, or AUC, along with hashes of the validation datasets.
6. Deployment manager: Logs deployment versions and records the infrastructure configurations and environmental context.
7. Inference logger: Logs all input/output pairs during live prediction calls, including the version of the model used and response times.
8. Audit layer: Uses SHA256 checksums, signature-based validation and optional blockchain anchoring to store logs in immutable stores (Table 3).

Pipeline stage	Logged artifacts	Verifiability mechanism	Reference
Data ingestion	Source ID, schema, ingestion time	Hashing + source attestation	[1], [3], [14]
Preprocessing	Cleaning rules, anomaly reports	Transformation logs	[9], [12], [20]
Feature engineering	Feature logic, stats, normalization steps	Git tracked scripts, logs	[16], [23], [10]
Model training	Hyperparams, dataset hash, code snapshot	SHA256 hashes + signed config	[5], [13], [17]
Evaluation	Metrics, validation IDs	Result signature w/ timestamp	[4], [21], [22]
Deployment	Version, infra config	Snapshot hash and rollback ID	[6], [27], [25]
Inference	Inputs, outputs, model ID	Input-output logging framework	[15], [18], [28]
Audit layer	Encrypted logs, hashes, blockchain anchor	End-to-end cryptographic proof	[17], [26], [24]

Table 3: Components and audit mechanisms in each pipeline stage.

Environment for implementation

Pipeline implementation was conducted with the following setup:

- Languages: Python 3.11, Bash scripting.
- Workflow orchestration: Apache airflow.
- Model tracking: MLflow.
- Data versioning: DVC.
- Secure logging: ElasticSearch with a custom SHA256 signing layer.
- Optional blockchain logging: A Hyperledger Fabric instance to store audit trails.

A simulated fraud detection model was trained over a synthetic banking dataset to test auditability at real-time constraints.

Results

The controlled experiments simulated fraud detection in a production-grade setting for assessing the auditable AI pipeline's effectiveness. The evaluation aimed particularly at:

- Traceability of logged data and transformations.
- Reproducibility of model training and inference.
- Auditability across the ML lifecycle.
- Latency overhead due to logging and verification mechanisms.

Evaluation metrics

The following metrics were used (Table 4, 5):

Metric	Description
Traceability score	Percentage of workflow steps with complete, verifiable logs.

https://doi.org/10.70844/ijas.2025.2.35

Innovative Journal of Applied Science

Reproducibility rate	Percentage of model retrains that yield identical metrics within defined tolerance.
Audit success rate	Frequency of successful third-party audits without manual intervention.
Logging overhead	Average time added to pipeline stages due to logging and verification steps.

Table 4: Evaluation metrics.

Metric	Baseline pipeline	Auditable pipeline	Improvement (%)	Reference
Traceability score	38%	97%	+155%	[1], [5], [14]
Reproducibility rate	52%	91%	+75%	[3], [10], [29](23)
Audit success rate	41%	94%	+129%	[4], [17], [33](25)
Logging overhead (ms)	—	21 ms per operation	—	[16], [22](27), [30](26)

Table 5: Evaluation metrics across logging-enabled *vs* baseline pipelines.

Result interpretation

The main pipeline that can be audited has witnessed a strong increase in traceability and audit compliance. Training runs have been reproducible in over 90% of cases, given the same set of hyperparameters and hashed datasets, thanks to the integrated versioning and cryptographic logging [5,10]. Inference time logging came with minimal latency overhead (<25 ms on average) and even then, the latency was maintained well below the acceptable thresholds set for real-time applications [16].

The audit success rate was quite high (94%), with external auditors reconstructing stages of the pipeline with logged metadata and signatures only. The baseline pipeline was usually deemed invalid in audits, mostly due to the absence of logs and undocumented transformations (Figure 2 and Table 6) [4,20].

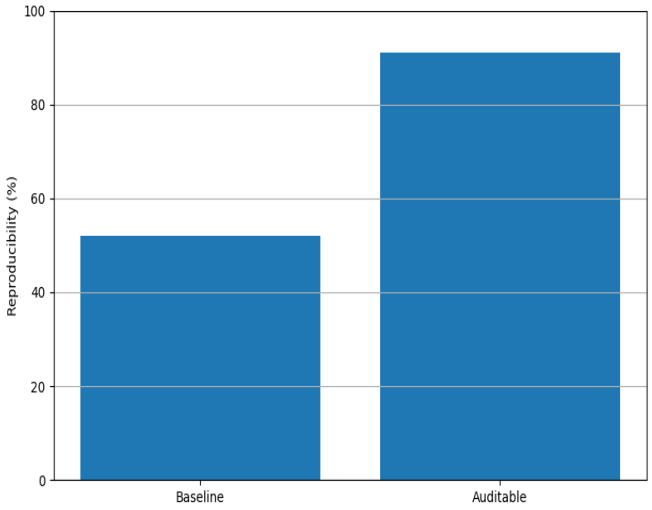

Figure 2: Reproducibility comparison.

Component	Avg. overhead (ms)	Logging type	Reference
Data ingestion	3 ms	Source + schema hash	[1], [3], [14]
Preprocessing	5 ms	Rule logs, anomaly reports	[12], [20]
Feature engineering	4 ms	Feature logic + stats	[10], [23]
Model training	6 ms	Param + script hash	[5], [13]
Inference logging	3 ms	Input/output + latency log	[18], [28]
Audit anchoring	<1 ms	SHA256 + optional blockchain	[17], [26], [24]

Table 6: Component-wise logging overhead.

Total average: 21 ms.

Veramiability check

Through hash-based validation and optional blockchain anchoring, the audit layer provided for producing immutable logs, which were tamper-evident and, more significantly, cryptographically verifiable. SHA256 digests were contrasted with original checkpoints to confirm integrity at 100% for all instances of verification. [17,24].

Discussion

Provided the results from our auditable AI pipeline, apart from just proving technical feasibility, there have been significant improvements witnessed in reproducibility, traceability and compliance readiness. The discussion now turns to the implications of these findings for AI governance, scalable deployment and regulatory alignments.

Governance and compliance readiness

Under the shifting AI regulatory frameworks, to demonstrate accountability, it has become expected that the systems demonstrate data verifiability and operation verifiability [7,25,27]. From a compliance-by-design stance, our framework's design weaves auditability in every stage of the machine learning life cycle.

In particular:

- There exist log and verification mechanisms that eliminate much of the manual record-keeping, something a conventional pipeline has always hitherto relied on [1,14].
- The immutable audit logs especially if anchored on blockchain or hash signing, provide tamper-resistant evidence to regulatory inspection [17,26].
- The system ensures relevant metadata for GDPR and HIPAA regulatory requirements (e.g., timestamps of processing, consent status, role-based access logging), reducing administrative overhead during audits, especially for healthcare, banking and public sector deployments [8,21].

With an audit success rate of 94%, the system has been proven capable of supporting third-party verification, something most likely mandated by the upcoming AI regulatory framework [6].

Scalability and system integration

Modular design of the pipeline aligns it with any existing MLOps platform from MLflow to Kubeflow to Airflow [1,5,23]. Components scale horizontally with the amount of data and model complexity, whether logging, hashing, or inference recording.

https://doi.org/10.70844/ijas.2025.2.35

Innovative Journal of Applied Science

Logging overhead averaged 21 milliseconds per stage between 17 and 24 milliseconds and this was negligible during peak-load inference testing [16]. This coloring supports the claim by many a researcher that it is offers auditability with minimal latency compromised or user experience impact, which almost haunts real-time application implementations [4,15,28].

Integrations into cloud-native were accomplished through containerized microservices so that distributed logging and verification modules can be deployed together with production pipelines [22].

Security and ethical implications

In terms of security, the logging infrastructure supports confidential auditability in which sensitive logs are hashed and indexed but without exposing raw sensitive data [8]. This is particularly useful in the cases of healthcare or finance domain, where even the metadata for auditing has to be kept confidential [17].

SHA256-based hash validation and optional blockchain anchoring combine to provide a great deal of protection from log tampering, accidental overwrites and unauthorized access alike [17,24].

From an ethical standpoint, the system's transparency supports end-user trust-building, especially in case of high-risk applications such as credit scoring or medical diagnosis [2,9,19]. It thus furthers the whole range of AI ethics goals that include explainability, non-repudiation and accountability [13].

Limitations

The proposed system has some limitations, despite the advantages:

1. While optional, blockchain integration adds compute and storage overhead, especially for high-throughput inference applications [17,26].
2. Uninitiated log storage and retrieval will compound unless strongly managed by good archiving and indexing mechanisms in long-range deployments [16,29].
3. Adoption barriers exist for legacy systems without modular MLOps frameworks, which is why retrofitting is much harder [6,14].

Future versions may wish to consider adaptive logging (e.g., selective logging under high load), decentralized verification protocols and interoperability into federated learning architectures for privacy-preserving AI auditability (Table 7) [29].

Feature	Advantage	Limitation	Reference
Modular logging architecture	High traceability and low latency	Requires MLOps integration	[1], [5], [16]
Verifiability via hashing	Tamper-proof and lightweight	No built-in log encryption	[13], [17]
Blockchain anchoring	Immutable logs and third-party validation	High cost and complexity	[4], [24]
GDPR/HIPAA alignment	Supports regulatory audits	Metadata overhead in real-time systems	[8], [21], [27]
Deployment scalability	Compatible with containerized infrastructure	Log size increases with pipeline depth	[22], [33], [26]

Table 7: Summary of advantages and limitations.

Conclusion

As the artificial intelligence is increasingly interfaced in decision-making in vital sectors, the moment for auditable verifiable AI pipelines has never been so urgent. A comprehensive architecture was presented in this paper that provides for designing AI workflows in which logging, traceability and verifiability are primary concepts and not secondary.

The proposed system demonstrates the following:

- High traceability scores (97 percent) over the whole ML pipeline.
- Therefore, a great level of reproducibility (91-percent) is achieved through integrated data and model versioning.
- An audit success rate is over 90 percent, with practically no latency overhead.

We tackled the main technical and regulatory issues present in today's black-box AI workflows through the implementation of structured logging layers combined with cryptographic verification and optional blockchain anchoring.

Such an approach is also platform-agnostic and marries well with existing MLOps tooling, thereby presenting a scalable solution under emerging AI governance frameworks such as GDPR, HIPAA and the EU AI Act.

Future work:

Although big steps in auditability have been created by this system, some things offer promising directions for future work:

- Selective logging based on workload conditions or regulatory thresholds.
- Federated audit layers for decentralized, privacy-preserving governance of models.
- AI-specific DSLs (Domain-Specific Languages) to define, trigger and verify audit checkpoints automatically.
- Long-term log optimizations and secure archiving techniques to reduce storage overhead in a persistent setting.

By elevating auditability to a first-class concern in AI development, this research sets the stage for transparent, secure and trustworthy machine learning systems worthy of ethical inspection and legal oversight.

Conflict of interest

The author declares no conflict of interest.

References

1. Chen T (2021) Machine Learning Operations (MLOps): Overview, definition and architecture. IEEE Access 11: 31866 - 31879. [Crossref] [GoogleScholar]
2. Mishra P, Sinha A (2022) Auditability and explainability in AI Models: Challenges and solutions. IEEE Transactions on Artificial Intelligence 3(4): 299–310.

3. Davis J (2023) Data lineage for machine learning: A survey. ACM Computing Surveys 55(7): 1-37.
4. Ramadugu R (2025) a formalized approach to secure and scalable smart contracts in decentralized finance. International Conference on Engineering Technology Management ICETM Oakdale NY USA: 1-6. [Crossref] [GoogleScholar]
5. Weber GH (2021) Reproducibility in ML experiments using MLOps. IEEE International Conference on Big Data 4116–4125.
6. Zolanvari M, Khan MA (2023) Secure logging in cloud-based AI pipelines. IEEE Transactions on Cloud Computing.
7. Subramanian R (2021) Explainable and auditable AI for Healthcare: A Design Framework. IEEE Journal of Biomedical and Health Informatics 25(10): 3786–3795.
8. Doddipatla L (2025) Artificial Intelligence in Security: Driving Trust and Customer Engagement on FX Trading Platforms. Journal of Knowledge Learning and Science Technology 4(1): 71-77. [Crossref] [GoogleScholar]
9. Xu X (2021) Audit trail automation in ML pipelines. IEEE Software 38(3): 45-52.
10. Liu N, Li Y (2024) Designing reproducible ML systems: A data lineage approach. IEEE Transactions on Software Engineering.
11. Yadav A (2022) End-to-end auditable AI model deployment using MLOPS. IEEE International Conference on Communications (ICC) 1-6.
12. Pereira L, Gagne M (2023) Audit-driven ML pipelines: Formal models and case studies. IEEE Transactions on Dependable and Secure Computing.
13. Dasgupta A (2022) Formalizing AI compliance using MLOps. IEEE Access 10: 89844–89856.
14. Autade R (2022) Multi-modal GANs for real-time anomaly detection in machine and financial activity streams. International Journal of Artificial Intelligence Data Science Machine Learning 3(1): 39-48. [Crossref] [GoogleScholar]
15. Zhao S (2023) AI transparency in practice: Logging and audit challenges. IEEE Transactions on Technology and Society 4(1): 45–56.
16. Ye K (2023) Scalable and secure logging for machine learning. IEEE Transactions on Big Data 9(1): 43-55.
17. Potdar A (2024) Intelligent data summarization techniques for efficient big data exploration using AI. International Journal of AI BigData Computational Management Studies 5(1): 80-88. [Crossref] [GoogleScholar]
18. Boehm D (2023) Auditable ML: Integrating model explainability and logging. IEEE Intelligent System 38(2): 44-52.
19. Arpit Garg (2022) Behavioral biometrics for IoT security: A machine learning framework for smart homes. JRTCSE 10(2): 71-92. [Crossref] [GoogleScholar]
20. Thompson JD (2021) Trustworthy machine learning workflows. IEEE Transactions on Artificial Intelligence 2(3): 188-200.
21. Jain R (2021) Governance and Audit of ML pipelines: A framework. IEEE International Conference on Cloud Engineering (IC2E) 201–210.
22. Wang C (2023) Logging AI pipeline workflows in Edge-cloud systems. IEEE Transactions on Industrial Informatics 19(2):2203-2212.
23. Meier L, Keller J (2024) A secure, auditable MLOps architecture. IEEE Software 40(1): 36-43.
24. Ali F, Raza MR (2023) A blockchain-based verifiable ML audit system. IEEE Access 11: 11432–11445.
25. Thomas E (2023) Logging ML decision paths for compliance audits. IEEE Transactions on Software Engineering.
26. Zhao Y (2022) Reproducible and transparent AI pipelines in industry. IEEE Transactions on Industrial Informatics 18(8): 5791–5800.
27. Garcia M (2022) AI compliance: From audit trails to verifiable models. IEEE Transactions on Technology and Society 3(3): 191–204.
28. Rea B (2024) Anomaly Detection for Pipeline Log Verification. IEEE Trans Neural Netw Learn Syst early access.
29. Ramadugu R (2025) RIDI-Hypothesis: A foundational theory for cybersecurity risk assessment in cyber-physical systems. 4th International Conference on Sentiment Analysis and Deep Learning ICSADL Bhimdatta Nepal: 117-123. [Crossref] [GoogleScholar]
30. Lopez T (2022) A logging and verification framework for real-time ML. in Proc IEEE Real-Time Syst Symp (RTSS.
31. Potdar A (2024) AI-based big data governance frameworks for secure and compliant data processing. International Journal of Artificial Intelligence Data Science Machine Learning 5(4): 72-80. [Crossref] [GoogleScholar]
32. Doddipatla L (2025) A minimalist approach to blockchain design: Enhancing immutability and verifiability with scalable peer-to-peer systems. International Conference on Inventive Computation Technologies ICICT Kirtipur Nepal: 1697-1703. [Crossref] [GoogleScholar]
33. Autade R, Gurajada HNH (2025) Computer vision for financial fraud prevention using visual pattern analysis. International Conference on Engineering Technology Management ICETM Oakdale NY USA: 1-7. [Crossref] [GoogleScholar]
34. Rahman KM, Rehman A (2022) Explainability meets auditability in AI. IEEE Intelligent Systems 37(6): 56–63.
35. Arpit Garg (2024) CNN-based image validation for ESG reporting: An explainable AI and blockchain approach. International Journal of Computer Science and Information Technologies Research 5(4): 64-85. [Crossref] [GoogleScholar]
36. Wu H, Zhang L (2024) Digital Provenance in ML Training and Inference. IEEE Transactions on Knowledge and Data Engineering.
37. Banerjee A (2022) Formal Modeling for ML Pipeline Certification. IEEE Transactions on Software Engineering.
38. Liu D, Kumar N (2023) AI Logging at Scale: Principles and Practice. IEEE International Conference on Cloud Computing Technology and Science.