

Evaluating Poisoning Resistance and Blockchain Provenance in Federated Intrusion Detection across Institutions

Srijib Samanta

Assistant Professor and Head of Department (HOD), Experience: More than 14 years Department of Bachelor of Computer Application (BCA) Tamralipta Institute of Management & Technology

Abstract:

Institutions can share intrusion detection models without pooling their network records, but federated learning remains vulnerable to corrupted training updates. Collaborative training also creates a need to reconstruct how a deployed model was produced and approved. This conceptual paper develops an architecture and evaluation protocol that examine poisoning resistance and auditability as separate outcomes. A focused review informs an institutional threat model, off-chain update screening, robust aggregation and permissioned records of model lineage. The protocol compares ordinary federated averaging, independently witnessed signed logs and blockchain-backed coordination under matched training conditions. Public intrusion detection datasets support controlled institutional partitions, which are distinguished from real deployments. The analysis shows why authenticated provenance cannot establish label correctness or update safety. It identifies targeted attack success, worst-institution detection performance, honest-client exclusion, evidence reconstruction and infrastructure cost as complementary evaluation endpoints. No training experiments or measured improvements are reported. The contribution is a controlled study design for determining whether shared ledger custody offers practical accountability beyond a strong signed-log alternative.

Keywords: federated learning; intrusion detection; model poisoning; permissioned blockchain; provenance; institutional cybersecurity.

1. Introduction

An institution sees attacks through its own network traffic, whereas a campaign may affect several organisations. Sharing knowledge could improve detection, but transferring traffic records creates confidentiality and access-control concerns. Federated learning addresses this difficulty by exchanging locally trained model updates rather than combining raw records (McMahan et al., 2017). For colleges and research organisations, participation is usually institutional and governed by an agreement.

Local data retention does not secure the learning process by itself. A compromised server, malicious insider or unreliable labelling process can produce harmful updates. Targeted poisoning may cause selected malicious traffic to be classified as benign while leaving overall accuracy relatively high (Bagdasaryan et al., 2020). Evaluation must therefore examine particular attack outcomes as well as average detection performance.

A second concern is accountability. Investigators may need to identify the parent model, preprocessing version, accepted contributions and approval decision behind a deployed detector. Blockchain-supported

federated intrusion detection already appears in earlier research (Preuveneers et al., 2018). The contribution proposed here is consequently an evaluation framework, rather than the first combination of these technologies.

1.1 Research problem and objectives

The research problem is to assess collaborative intrusion detection when institutions have different traffic distributions and some contributors may be malicious. The assessment must distinguish defenses that change the learned model from records that preserve evidence about training and deployment.

The study has four objectives: define the adversary and trust boundaries; specify an institutional learning and provenance architecture; design reproducible poisoning and heterogeneity tests; and compare blockchain-backed accountability with a signed-log system under equal evidence-retention conditions.

1.2 Research questions and propositions

Research question	Testable proposition
RQ1 Does provenance alone change poisoning outcomes?	P1 Identical updates and deterministic decisions produce the same model regardless of audit storage.
RQ2 How do defenses behave under institutional diversity?	P2 Robust processing may reduce poisoning while excluding honest minority contributions.
RQ3 Does a ledger improve independent reconstruction?	P3 Shared records may resist selected history alterations within explicit fault assumptions.
RQ4 What operational costs follow?	P4 Ledger coordination adds infrastructure and evidence-management costs that require measurement.

2 Related research and evaluation gap

2.1 Federated optimisation and poisoning

McMahan et al. (2017) establish iterative model averaging as a practical federated learning approach. Li et al. (2020) address statistical and systems heterogeneity through FedProx. Together, these studies support clean heterogeneous controls: weak performance can arise from differences between institutions even when no participant is malicious.

Bagdasaryan et al. (2020) demonstrate targeted compromise through model replacement. Fung et al. (2020) examine coordinated contributors in Sybil settings. These threats motivate separate tests for corrupted data, manipulated updates and coordinated influence. Institutional enrolment can restrict duplicate identities, but genuine enrolled organisations may still collude.

Krum, coordinate-wise median and trimmed mean offer approaches to Byzantine-robust learning (Blanchard et al., 2017; Yin et al., 2018). Their assumptions must be checked before use. Non-IID institutional data can undermine the usefulness of treating distance from the majority as evidence of malice. FLTrust additionally depends on a protected root dataset whose representativeness requires scrutiny (Cao et al., 2021).

2.2 Provenance and confidentiality

Preuveneers et al. (2018) investigate chained records in federated anomaly detection, while Mehammed et al. (2025) develop blockchain-audited federated learning. The W3C provenance model provides a complementary vocabulary of entities, activities and agents (Moreau & Missier, 2013). Model versions are entities; training and aggregation are activities; institutions and reviewers are responsible agents.

Provenance describes recorded events and their supporting artifacts. It does not establish that labels are correct or that a submitted update is safe. Similarly, data locality is not a complete privacy guarantee: gradients may reveal training information under particular conditions (Zhu et al., 2019). Secure aggregation can conceal individual updates (Bonawitz et al., 2017), but ordinary secure summation conflicts with screening that requires those individual updates to be visible.

2.3 Evaluation gap

This focused review is purposive rather than systematic. Its contribution is a comparison that separates statistical defense from evidence infrastructure, includes a strong signed-log alternative, measures honest-client exclusion, and tests whether retained evidence supports independent reconstruction. These features make a ledger's incremental value assessable without treating transaction counts or aggregate accuracy as proof of security.

3 Research methodology and threat model

The study is an exploratory design paper based on primary research and technical documentation. It combines literature-informed requirements, a threat model, an architecture and a proposed experimental protocol. It reports no interviews, institutional traffic collection or executed training experiments. The evaluation unit will be an independently seeded consortium run. Individual institutions and repeated rounds within a run are nested observations, not independent replications.

The threat model separates external network attacks, which the detector should recognise, from internal training attacks, which seek to corrupt it. Malicious clients may change local labels, manipulate updates or coordinate within a declared per-round budget. That budget is reported as both a fraction of active institutions and the influence weight available before defensive processing.

Evidence attacks form a separate family. They include artifact modification, replay, missing reports and inconsistent histories. A stolen signing key is also considered: a valid signature identifies the credential used, but does not prove who controlled it. Malicious ledger operators are tested only where the selected consensus configuration supplies a corresponding fault model.

Table 1 Threats and evaluation evidence

Threat	Proposed control	Required evidence
Poisoned data or updates	Screening and robust aggregation	Targeted attack success and detection deterioration
Replayed or wrong-parent update	Round and parent-model checks	Explicit acceptance and rejection records
Altered stored artifact	Commitment and signature verification	Mismatch against retrieved artifact bytes
Conflicting histories	Independent witnesses or ledger replicas	Cross-reviewer consistency checks
Lost evidence or unavailable nodes	Retention and availability policies	Retrieval failures and incomplete rounds
Gradient or metadata exposure	Restricted access and minimal records	Access audit and separate privacy assessment

Initial experiments assume a common feature contract and a protected validation reference. Dedicated stress tests then relax these assumptions. Full compromise of the validation service and faults beyond the declared consensus threshold are outside the primary resistance claim. Their operational consequences should still be documented.

4 Institutional architecture

4.1 Membership and feature contracts

Each institution receives an organisational identity with defined training, review and administration roles. Enrolment, certificate renewal, revocation and key rotation follow a consortium agreement. Creating extra local processes does not increase an organisation's permitted influence. Historical records remain available after membership changes.

A versioned feature contract defines feature names, order, units, missing-value treatment and label mappings. Every update identifies its parent model and preprocessing version. Scaling and other fitted transformations use training data only. The initial evaluation uses a shared model architecture; personalisation remains a later extension so that aggregation comparisons retain a clear interpretation.

4.2 Evidence storage and verification

Traffic records, model-update arrays and full validation reports remain in encrypted off-chain repositories. The ledger records organisational identity, round, parent-model commitment, update commitment, preprocessing version, report reference, acceptance or rejection reason, and resulting model commitment. Rejected submissions retain an explicit status instead of disappearing from the trail. Commitments require a specified collision-resistant hash and canonical serialization, including tensor names, shapes, order and numerical representation. An authorised verifier retrieves artifacts and checks their bytes. Sensitive low-entropy metadata may require nonce-based commitments with restricted nonce access; this measure is not a complete privacy guarantee. Validators recompute numerical properties rather than trusting client-reported values.

Table 2 Learning decisions and evidence boundaries

Processing layer	Main action	Evidence boundary
Institutional training	Retain traffic locally and submit signed update	Parent model and feature-contract version
Off-chain validation	Check validity, screen and aggregate	Protected updates and complete test reports
Shared evidence records	Commit event references and dispositions	Witnessed signed log or permissioned ledger
Release and local inference	Approve exact model and deploy locally	Approval record linked to model commitment

The validation service can inspect individual updates in this configuration. Confidentiality from that service is therefore not claimed. Ledger endorsement verifies required authorisations and event consistency; it cannot establish the scientific adequacy of a validation test. Inference remains local and does not require a transaction for each network flow.

4 Proposed institutional architecture continued

4.3 Screening and aggregation

For K institutions, let $L_k(w)$ denote local loss and q_k a declared nonnegative weight. The shared training objective is:

$$L(w) = \sum_k q_k L_k(w), \quad q_k \geq 0, \quad \sum_k q_k = 1. \quad (1)$$

Equal institutional weighting uses $q_k = 1/K$. A separate sample-weighted baseline uses training counts capped under consortium policy. Signed counts remain assertions unless independently checked. Comparable audit configurations retain identical weighting choices.

For submitted update $\Delta_{k,t}$, admission checks identity, round freshness, parent model, dimensions and finite values. Optional off-chain screening checks reference-set deterioration and update magnitude. Accepted updates are clipped to a predeclared norm bound C :

$$\tilde{\Delta}_{k,t} = \Delta_{k,t} \min(1, C / \|\Delta_{k,t}\|_2), \text{ with } \tilde{\Delta}_{k,t} = 0 \text{ if } \Delta_{k,t} = 0. \quad (2)$$

For accepted set A_t and aggregation rule R , the next candidate is:

$$w_{t+1} = w_t + R(\{\tilde{\Delta}_{k,t} : k \in A_t\}). \quad (3)$$

Clipping limits magnitude but does not remove every harmful direction. Coordinate-wise median and trimmed mean are evaluated separately. Trimming levels and participant minima must be feasible. Optional Krum and FLTrust comparisons retain their original admissibility and root-data requirements. If too few updates remain, the round fails and the last approved model stays active; these failures count toward availability outcomes.

4.4 Approval recovery and governance

Aggregation creates a candidate rather than an automatically approved deployment. Independent reviewers check its tests and lineage against a declared release policy. The release record identifies exact model bytes and the evidence bundle. Quarantine or rejection includes a reason and a review route for honest but unusual contributors.

Rollback selects a previously approved checkpoint. It cannot erase poisoning already propagated through later models; recovery may require retraining from an earlier clean checkpoint. Time, success and evidence requirements for this process are measured.

Hyperledger Fabric is a possible implementation (Androulaki et al., 2018). Raft supports crash-fault tolerance, whereas BFT ordering addresses a bounded set of malicious orderers (Hyperledger Fabric, n.d.). A four-orderer, one-fault experiment requires genuinely independent administration and the selected implementation's assumptions. Ordering consensus and institutional release endorsement remain separate controls.

5 Proposed evaluation protocol

5.1 Benchmarks and institutional partitions

CIC-IDS2017 and UNSW-NB15 provide documented public benchmarks (Canadian Institute for Cybersecurity, n.d.; UNSW, n.d.). Each is evaluated separately because its features and attack taxonomy differ. Their age and traffic-generation setting limit claims about present institutional networks. Simulated clients are benchmark partitions, not evidence of a real consortium deployment.

Training, validation and testing are separated before assigning clients. Identifiable duplicate or near-duplicate groups remain within one split. Temporal or capture-session separation is preferred; grouped splitting is used when suitable timestamps are unavailable. Test records never determine preprocessing

parameters, clipping bounds, screening thresholds or hyperparameters. Feature exclusions are justified by leakage risk and deployment availability and recorded in the manifest.

The protocol proposes $K = 5, 10$ and 20 simulated institutions under approximately IID, label-skew and quantity-skew partitions. Dirichlet label allocations with $\alpha = 0.1, 0.5$ and 1.0 offer controlled skew levels, subject to documented minimum sample sizes. Capture-group or unseen-time tests provide complementary challenges where metadata permits. Client-level class coverage and absent classes are reported explicitly.

5.2 Model and training controls

An initial tabular model uses two hidden layers of 64 and 32 units. Starting choices are Adam, learning rate 0.001 , batch size 128 , three local epochs and a maximum of 100 communication rounds. These are proposed pilot settings rather than established optima. Binary detection is evaluated first; multiclass outcomes are analysed separately under a shared label mapping.

A clean validation pilot sets a small hyperparameter grid, screening rules and clipping bounds. These choices are frozen before attack evaluation. Test thresholds are not adjusted afterwards to improve results. Local-only training provides a deployment reference. A pooled-data model may supply an experimental reference where licensing permits, without implying that institutions should pool their operational records.

5.3 Matched comparisons

Within each comparison, systems share feature schemas, model initialisation, participation schedules, partitions, attacks and test sets. Signed logs include cryptographic signatures and independently retained witness checkpoints. Both evidence systems receive equal artifact-retention budgets and equivalent reviewer access. Comparing a ledger with an unsecured text file would overstate its practical contribution.

5 Proposed evaluation protocol continued

Table 3 Configurations separating model defense from audit infrastructure

Code	Learning configuration	Evidence infrastructure
B0	Federated averaging and input-validity checks	Ordinary coordinator records
B1	Identical updates and decisions to B0	Signed append-only log with independent witnesses
B2	Identical updates and decisions to B1	Permissioned ledger with matched artifact retention
B3	Screening, clipping and one robust aggregation rule	Signed log with independent witnesses
B4	Identical defense and decisions to B3	Permissioned ledger with matched artifact retention

Fixed-update replay first isolates evidence infrastructure: B1 versus B2 and B3 versus B4 retain the same interventions. A separate experiment allows governance to act on evidence under equal rules and investigation resources. Any prediction benefit in that experiment is traced to changed action or timing, rather than directly to the storage mechanism.

5.4 Adversarial and infrastructure scenarios

Malicious-client budgets are $0\%, 10\%, 20\%$ and 40% , restricted to feasible client counts and algorithm assumptions. Integer rounding and effective adversarial weight are disclosed. Clean runs measure the

cost of each defense before poisoning is introduced. Exceeding an algorithm's admissible fault budget is identified as a stress test, not treated as a violation of its theorem.

Poisoning scenarios include label corruption, constrained update manipulation and targeted misclassification. Targeted perturbations must represent network-valid behaviour and use a predefined eligible test set. Arbitrary changes to derived flow statistics can create impossible records. Adaptive attackers know the screening policy, so resistance is not inferred solely from rejecting a simple fixed perturbation.

Evidence scenarios introduce altered artifacts, wrong-parent submissions, repeated messages, missing reports and inconsistent histories. Validly signed poisoned updates are included because correct provenance can coexist with failed statistical defense. Reviewers reconstruct events against evaluator-held scenario ground truth.

Availability scenarios include client dropout, validator loss, ordering delay and unavailable off-chain storage. Receipt acknowledgment is distinguished from confirmed commitment. Timeouts, incomplete rounds and retained stale models remain visible in operational results, even where they do not directly change classification accuracy.

5 Proposed evaluation protocol continued

Table 4 Outcomes and operational definitions

Outcome	Measure and interpretation
Detection utility	Macro-F1, class recall and benign false-positive rate, reported globally and per institution, including the worst institution.
Targeted compromise	Eligible attacked malicious examples predicted benign divided by the predefined eligible attacked set.
Poisoning deterioration	Clean-minus-attacked performance under paired settings and equal budgets.
Honest-client exclusion	Rejected honest updates divided by eligible honest submissions, stratified by institution and partition.
Evidence completeness	Required evidence items retrieved and verified divided by all required items; missing artifacts count as failures.
Reconstruction success	Correct reconstruction of parent model, included contributions and release decision against hidden scenario ground truth.
Operational cost and recovery	Round duration, commit latency, traffic, CPU, storage, failed rounds, and time to an approved uncontaminated checkpoint.

Benign false-positive rate is $FP/(FP + TN)$, with benign traffic as the negative class. The targeted-attack denominator is fixed before comparison. Macro-F1 treatment of absent classes is predeclared. Overall accuracy is secondary because frequent benign observations can conceal rare attack failures. Reviewers distinguish evidence of a credentialed submission from an unsupported attribution of malicious intent.

5.5 Statistical analysis and reproducibility

The initial plan proposes ten independent seeds per feasible condition. A pilot assesses variability and cost before the final replication count is registered. Paired partitions and seeds support absolute difference estimates with run-level bootstrap confidence intervals. Repeated rounds and individual flows are not counted as independent experimental replications. Multiple-comparison families and corrections are declared if formal tests are used.

Primary endpoints are targeted attack success, worst-institution macro-F1, honest-client exclusion and reconstruction success. Operational thresholds accompany statistical uncertainty. Reproducibility requires dataset hashes, split manifests, code, seeds, model settings, attack budgets, defense parameters and machine specifications. Audit replication additionally requires event schemas, consensus settings, endorsement rules and retention policies. Byte-verified deterministic replay is distinguished from a new run with similar scores.

6 Conceptual analysis and discussion

The architecture supports four deductions. First, matching a commitment verifies consistency with registered bytes, not the correctness of labels or the safety of an update. Detecting later alteration requires an authentic commitment, available artifacts and actual verification. Lost evidence, compromised original inputs and stolen keys remain distinct failures.

Second, identical update tensors, deterministic aggregation and unchanged interventions produce an identical model trajectory. Audit storage supplies no additional term in Equation 3. Where numerical execution is nondeterministic, random streams and execution conditions must also be controlled. Any ledger-associated performance gain therefore requires an identified pathway, such as earlier quarantine or more reliable release review.

Third, institutional diversity can resemble suspicious behaviour. A specialised laboratory may submit an honest update unlike the majority. An attacker can also preserve ordinary update magnitude while damaging a narrow decision boundary. Per-institution performance, clean heterogeneous controls and honest-client exclusion are needed to distinguish these cases.

Fourth, an audit trail is useful only when supporting evidence is retrievable. Replicated commitments cannot reconstruct missing update arrays or reports. Evaluation should test usable evidence bundles rather than transaction volume. A ledger is most plausible where institutions need shared custody and do not accept a sole historian; witnessed signed logs may otherwise be sufficient.

7 Ethics limitations and future research

Initial work uses public benchmarks under their usage conditions and confines adversarial tests to an isolated environment. A later institutional pilot requires documented authorisation, restricted access, retention controls and appropriate review. Raw payloads, credentials and personal identifiers are excluded from shared records. Participation timing and organisational metadata may still disclose sensitive information.

The principal limitation is the absence of an evaluated prototype. Detection gains, infrastructure costs and institutional acceptance remain unmeasured. Benchmark partitions simplify operational diversity; the focused review is not exhaustive. Individual-update inspection creates a trust boundary at the validator. Future work should implement the matched comparators, test adaptive attacks and evidence loss, then assess drift, analyst usefulness and privacy-preserving validation in an authorised pilot. Funding, conflicts, assistance and author contributions require accurate journal-specific declarations.

8 Conclusion

Poisoning resistance and auditability require separate evidence. Robust processing may limit harmful contributions, while verifiable records support investigation and recovery within explicit trust assumptions. The proposed protocol tests both against a strong signed-log comparator, retaining

institutional diversity, evidence availability and operating cost as central criteria. Quantitative claims require implementation and independent evaluation.

References

1. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., ... Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. *Proceedings of the Thirteenth EuroSys Conference*, Article 30, 1–15. <https://doi.org/10.1145/3190508.3190538>
2. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. *Proceedings of Machine Learning Research*, 108, 2938–2948. <https://proceedings.mlr.press/v108/bagdasaryan20a.html>
3. Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30. https://papers.neurips.cc/paper_files/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html
4. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191. <https://doi.org/10.1145/3133956.3133982>
5. Canadian Institute for Cybersecurity. (n.d.). Intrusion detection evaluation dataset (CIC-IDS2017). University of New Brunswick. <https://www.unb.ca/cic/datasets/ids-2017.html>
6. Cao, X., Fang, M., Liu, J., & Gong, N. Z. (2021). FLTrust: Byzantine-robust federated learning via trust bootstrapping. *Proceedings of the Network and Distributed System Security Symposium*. <https://doi.org/10.14722/ndss.2021.24434>
7. Fung, C., Yoon, C. J. M., & Beschastnikh, I. (2020). The limitations of federated learning in Sybil settings. *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses*, 301–316. <https://www.usenix.org/conference/raid2020/presentation/fung>
8. Hyperledger Fabric. (n.d.). The ordering service. *Hyperledger Fabric documentation*. Retrieved October 14, 2026, from https://hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html
9. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2, 429–450. https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html
10. McMahan, B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of Machine Learning Research*, 54, 1273–1282. <https://proceedings.mlr.press/v54/mcmahan17a.html>
11. Mehammed, S., Bewuketu, G., Getaneh, D., Alam, M. N., Khan, S., & Alhayan, F. (2025). Blockchain-audited federated learning: Securing data and model updates with on-chain provenance. *IET Software*, 2025(1), Article 6670439. <https://doi.org/10.1049/sfw2/6670439>

12. Moreau, L., & Missier, P. (Eds.). (2013). PROV-DM: The PROV data model. World Wide Web Consortium. <https://www.w3.org/TR/prov-dm/>
13. Preuveneers, D., Rimmer, V., Tsingenopoulos, I., Spooren, J., Joosen, W., & Ilie-Zudor, E. (2018). Chained anomaly detection models for federated learning: An intrusion detection case study. *Applied Sciences*, 8(12), Article 2663. <https://doi.org/10.3390/app8122663>
14. UNSW. (n.d.). The UNSW-NB15 dataset. UNSW Research. <https://research.unsw.edu.au/projects/unsw-nb15-dataset>
15. Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *Proceedings of Machine Learning Research*, 80, 5650–5659. <https://proceedings.mlr.press/v80/yin18a.html>
16. Zhu, L., Liu, Z., & Han, S. (2019). Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32. https://papers.neurips.cc/paper_files/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html