Blockchain-Auditable Disaster Image Analysis for Trustworthy Damage Assessment

A Hash-Chained Provenance Ledger with Tile-Level Merkle Commitments for Flood and Cyclone Damage Mapping

[1]Pravat Satpathy, [2]Saswat Swain, [3]Sonia Mohapatra
[1]*Researcher*, [2]*Ph.D Scholar*, [3]*Researcher*
[1,2,3]*Research and Development Division,*
[1,2,3]*Tekstein Solutions Private Limited, Bhubaneswar, Odisha, India*
[1,2,3]analytics.pravat@gmail.com, saswat.mck@gmail.com, smctc98@gmail.com

Abstract*—Flood extents and building-damage maps derived from satellite and aerial imagery now feed relief allocation, insurance settlement and reconstruction funding, yet the processing history behind a published map is rarely recorded in a form that a third party can check. This paper presents a reproducible damage-assessment pipeline in which every artefact, from the archived source scenes through each intermediate product to analyst revisions of the final map, is committed to an append-only, hash-chained provenance ledger. Each commitment binds a Merkle root computed over fixed image tiles, a digest of the source code of the stage that produced it, its parameters and the commitments of its parents, and the ledger head is periodically anchored with an independent witness. The processing chain combines relative radiometric normalisation on pseudo-invariant pixels, median filtering, a water-index change rule and object-based building-damage scoring. It was evaluated on 180 synthetic bi-temporal scenes with exact ground truth. At and above 10 dB per-band signal-to-noise ratio the chain reached a damage F1 score between 0.958 and 0.981 against 0.909 to 0.955 for a band-difference baseline; below 8 dB it failed and the simpler baseline was better, a limitation reported rather than tuned away. Every local tampering scenario tested was detected and localised to the altered tile or record, whereas a full rewrite of the ledger by its operator was detectable only up to the most recent witness anchor, in close agreement with a derived exposure model. Re-execution on a different floating-point precision reproduced the class maps exactly but not the floating-point intermediates, even after quantisation. Provenance capture added about one fifth to processing time. The results indicate that auditable damage assessment is mainly a problem of canonical commitment and anchoring discipline rather than of consensus technology.

***Index Terms*—Blockchain, change detection, damage assessment, data provenance, disaster response, flood mapping, hash chain, Merkle tree, remote sensing, reproducibility, tamper evidence.**

I. Introduction

Within hours of a major flood or cyclone landfall, space agencies and rapid-mapping services release delineation and grading maps derived from optical and radar imagery [1], [2]. These products are no longer only situational-awareness aids. They are used to prioritise rescue sectors, to size relief transfers, to trigger parametric insurance and to justify reconstruction budgets, and population-scale exposure studies increasingly rely on archives of such maps [3]. When a map carries financial or legal weight, the question of how it was produced becomes as important as its accuracy: which scenes were used, which algorithm version and thresholds were applied, whether an analyst edited the result, and whether the file now on a portal is the file that was issued.

Current practice answers these questions through metadata sidecars, version-control repositories and institutional trust. None of these lets an outside party verify, after the fact, that an archived map and its lineage have not been altered, and none distinguishes an honest re-processing from a silent one. Blockchain platforms are frequently proposed as the remedy, but such proposals do not always separate the property that is actually needed, tamper evidence of a processing record, from the machinery that is often attached to it, such as consensus networks or tokens. A hash-chained log, the core data structure of a blockchain, already provides tamper evidence for every record except the most recent ones, provided that its head is periodically witnessed by a party the operator does not control [4]–[6].

This work treats auditable damage assessment as an engineering problem with three parts: a processing chain whose stages are pure functions of their inputs and parameters, a canonical commitment scheme for the image artefacts those stages exchange, and a ledger model whose integrity guarantees can be stated and measured. The contributions are as follows.

1) A provenance schema that commits source scenes, every intermediate product, the source-code digest and the parameters of each stage, and analyst revisions with explicit parent pointers, recovering all 17 provenance elements defined for one assessment.
2) Tile-level Merkle commitments over image artefacts, which detect any pixel alteration and localise it to the altered tiles, with an analysis of the tile-size trade-off.
3) A closed-form exposure model for storage-level ledger rewriting under periodic external anchoring, validated by Monte Carlo simulation.
4) A quantitative account of re-execution verification under floating-point variation, showing that digest quantisation does not make floating-point intermediates reproducible and that class maps are the appropriate unit for bit-exact re-verification.
5) A complete simulation study of the damage-mapping chain, including ablations and the operating range in which it is inferior to a simple baseline.

II. Related Work

Unsupervised change detection for disaster mapping has a long lineage. Difference and ratio images thresholded by histogram methods [7], [8] remain the operational default because they need no training data, while statistically modelled variants such as the iteratively reweighted multivariate alteration detector [9] improve robustness to radiometric differences between acquisitions. Water indices such as the normalised difference water index (NDWI) [10] give a physically interpretable flood rule for optical data, and benchmark collections such as Sen1Floods11 [11] and xBD [12] have enabled supervised flood and building-damage models. In all of this work the output is a map; the record of how it was produced is outside the scope.

Tamper-evident logging is equally mature. Time-stamping by hash linking [4], Merkle authentication trees [5], history trees for efficient audit [6] and certificate transparency logs [13] establish that integrity of an append-only record reduces to collision resistance plus external witnessing of the log head. Permissioned ledgers such as Hyperledger Fabric [14] add replication and access control, and ProvChain [15] showed how cloud-file operations can be recorded on a blockchain. The W3C PROV model [16] standardises the vocabulary of entities, activities and agents. What remains open for damage assessment is the binding between these two bodies of work: commitments that remain meaningful for large image arrays, record schemas that cover the parameters and code that determine a map, and measurements of what such a design does and does not detect. Table I summarises the gap.

TABLE I
Representative Prior Work and Gap Addressed

Work	Focus	Method	Limitation here
[7], [8]	Change detection	Difference or ratio image, automatic threshold	No processing record
[9]	Change detection	Iteratively reweighted MAD	No processing record
[11], [12]	Flood, damage data	Benchmark datasets	Accuracy only, no lineage
[6], [13]	Tamper-evident logs	Hash chain, Merkle audit	Generic records, no image tiles
[15]	Cloud provenance	Blockchain-recorded file operations	File-level hashes, no stage semantics
This work	Auditable damage maps	Stage commitments, tile Merkle, anchoring	Synthetic evaluation only

III. Problem Formulation and Threat Model

An assessment A maps a pre-event scene X_0 and a post-event scene X_1 to a class map D through a chain of stages $f_1, \ldots, f_m$, each with parameters θ_k and implementation c_k. Analysts may later issue revisions D', D'' of the map. The auditor must be able to answer, for any issued map: which source scenes were used, which code and parameters produced each intermediate, who revised the map and why, whether any artefact changed after it was recorded, and whether the map can be re-derived from the recorded inputs.

Three adversaries are considered. An *artefact adversary* edits stored images or maps, for example to hide damage in a sector. A *record adversary* edits, deletes or reorders individual ledger records. A *storage adversary*, typically the ledger operator, can rewrite the entire stored ledger and recompute every hash, but cannot alter digests already published to an independent witness. Compromise of the processing host before commitment, collusion between operator and witness, and attacks on SHA-256 are out of scope. Practical constraints are that scenes are large, so committing must not dominate processing time, and that re-execution may happen years later on different hardware.

IV. Proposed Methodology

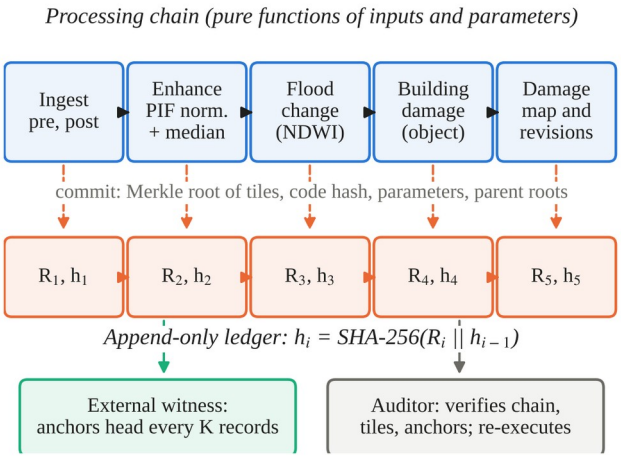


Fig. 1. System architecture. Each stage of the processing chain commits its output, code digest, parameters and parent commitments to an append-only hash-chained ledger; an external witness holds the ledger head every K records; an auditor verifies the chain, the tile commitments and the anchors and may re-execute the chain.

A. Acquisition Model and Enhancement

Each band b of the post-event scene is modelled as

$$y_b = g_b x_b + o_b + n_b, \tag{1}$$

where x_b is the reflectance seen in the pre-event geometry, g_b and o_b are radiometric gain and offset differences between acquisitions, and n_b is zero-mean noise. Enhancement first normalises the post-event scene to the pre-event scene on pseudo-invariant pixels (PIFs) [17], selected as the fraction ρ of pixels with the smallest residual and refined over three iterations starting from the identity mapping. The mapping is fitted by moment matching,

$$\hat{a}_b = s_x / s_y, \quad \hat{c}_b = m_x - \hat{a}_b m_y, \tag{2}$$

with m and s the PIF means and standard deviations. Ordinary least squares was used in the first implementation and is retained as an ablation. Its slope is biased toward zero when the regressor is noisy [18],

$$\mathrm{E}[\hat{a}_{\mathrm{OLS}}] \approx \lambda / g_b, \quad \lambda = g_b^2 \sigma_x^2 / (g_b^2 \sigma_x^2 + \sigma_n^2), \tag{3}$$

which compresses the dynamic range of the normalised scene and, as Section VII shows, removed flood detection entirely up to 10 dB. A 3 × 3 median filter is then applied to both scenes.

B. Flood Change and Building Damage

Water is identified with the NDWI [10],

$$W = (G - N) / (|G| + |N| + \varepsilon), \tag{4}$$

where G and N are the green and near-infrared bands. New flood water is

$$F = [W_1 > \tau] \wedge \neg [W_0 > \tau], \quad \tau = \max(\tau_{\mathrm{Otsu}}, 0), \tag{5}$$

with τ_{Otsu} the Otsu threshold [8] of the pooled pre- and post-event NDWI. The floor at zero is the McFeeters criterion. Without it, bright roofs with NDWI near −0.06 fell on the water side of the pooled threshold, so flooded buildings were discarded as pre-existing water; this defect was found during development and the floor was added. F is cleaned by an opening and a closing with a unit disc and removal of components below 25 pixels.

Building footprints are extracted from the pre-event green band by Otsu thresholding. For each footprint, a damage score is computed on its interior, eroded by half the window width and kept clear of flood pixels to avoid mixing at edges,

$$S = \mathrm{median}_{\Omega}(|\mu_1 - \mu_0| + \sigma_1 - \sigma_0), \tag{6}$$

where μ and σ are 5×5 local means and standard deviations of the near-infrared band and Ω is the footprint interior. Rubble lowers the mean and raises local texture, so both terms increase. Footprints more than half covered by F are assigned to the flood class; the remainder are labelled damaged when S exceeds the Otsu threshold of all footprint scores, with a floor of 0.02. The class map D takes the value 1 for new flood water and 2 for damaged buildings.

C. Canonical Commitments and the Ledger

An artefact is serialised canonically as a header (data type, shape, quantisation step) followed by its little-endian row-major bytes. It is divided into $T \times T$ tiles, each leaf being $\ell_j = \mathrm{H}(\text{coordinates} \,\|\, \text{tile bytes})$, and committed by the Merkle root

$$r = \mathrm{MerkleRoot}(\ell_1, \ldots, \ell_J), \tag{7}$$

with domain-separated internal nodes [5]. A record R_i holds its index, kind (ingest, process or revise), payload and the previous record hash, and

$$h_i = \mathrm{H}(\mathrm{canonical}(R_i) \,\|\, h_{i-1}), \quad h_0 = 0^{256}, \tag{8}$$

with H the SHA-256 function. A process record contains the stage name, the digest of the stage source code, its parameters, the data-dependent threshold it chose, the parent roots and the output root. A revise record contains the parent map root, the analyst identity, a reason and the new root, so earlier versions are never overwritten. Every K records the head h_i is sent to an external witness, a role that a public blockchain transaction, a transparency log [13] or a second agency can fill.

D. Integrity Properties

Proposition 1 (tile localisation). If an artefact is altered on a set of tiles S, its recomputed root differs from the committed root if and only if S is non-empty, and the mismatching leaves are exactly S, unless a second preimage of SHA-256 is found. Each leaf binds its coordinates and contents, so unaltered tiles reproduce their leaves, altered tiles cannot without a second preimage, and the root binds the ordered leaves.

Proposition 2 (anchoring exposure). A storage adversary can alter record j without detection if and only if no anchor index at or after j existed when the attack occurred. With an anchor every K records and ledger length n at the time of attack, the exposed records are the last (n mod K). If n is uniform on $\{1, \ldots, L\}$ and j is uniform on the existing records, the detection probability is

$$P_{\mathrm{det}}(K) = 1 - (1/L) \sum_{n=1}^{L} (n \bmod K)/n. \tag{9}$$

Without anchoring $P_{\mathrm{det}} = 0$: a hash chain alone gives no protection against its operator.

Proposition 3 (quantised digests under perturbation). Let a real array be re-computed with element errors δ_i smaller than the quantisation step q, with sub-grid positions uniformly distributed. Element i changes its quantised value with probability $|\delta_i|/q$, so the digests agree with probability

$$P_{\mathrm{match}} = \prod_i (1 - |\delta_i|/q) \approx \exp(-\textstyle\sum_i |\delta_i|/q). \tag{10}$$

Because the sum grows with the number of elements, quantisation can make a large floating-point array reproducible only if q far exceeds N times the mean error, at which point it no longer commits to the values meaningfully.

E. Audit Completeness

For an assessment the required provenance set E_{req} has 17 elements: two source commitments; code digest, parameters and output for each of four stages; and parent, analyst and reason for a revision. For a logging policy recording E_{rec},

$$A = |E_{\mathrm{rec}} \cap E_{\mathrm{req}}| / |E_{\mathrm{req}}|. \tag{11}$$

A is measured by querying the records that each policy actually writes. It is a coverage measure relative to a requirement list defined by the authors, not evidence that the list is sufficient for every legal setting.

V. Algorithm and Complexity

Algorithm 1: Auditable damage assessment

Input: scenes X_0, X_1; parameters Θ; tile size T; anchor interval K
for X in (X_0, X_1): append ingest record with root of X
($\tilde{X_0}$, $\tilde{X_1}$) ← Enhance(X_0, X_1; θ_e) ▷ Eqs. (2), median
F ← Clean(FloodChange($\tilde{X_0}$, $\tilde{X_1}$)) ▷ Eqs. (4)–(5)
B ← BuildingDamage($\tilde{X_0}$, $\tilde{X_1}$, F; θ_b) ▷ Eq. (6)
D ← ClassMap(F, B)
for each stage k: append process record {H(c_k), θ_k, parent roots, root of output}
on analyst edit D': append revise record {root(D), analyst, reason, root(D')}
after each append: if (i + 1) mod K = 0, send h_i to witness
Audit: verify links, record hashes and anchors; recompute tile leaves of
stored artefacts; report mismatching tiles; optionally re-execute and compare D

For an image of N pixels, enhancement costs $O(N \cdot w^2)$ with median window w, Otsu thresholding $O(N + b)$ for b histogram bins and the object stage $O(N)$. Committing an artefact costs $O(N)$ hashing plus $O(N/T^2)$ internal Merkle nodes; appending a record costs $O(1)$ and verifying a ledger of L records $O(L)$. Storage is about 0.5 kB per record plus 32 bytes per tile leaf, kept off-ledger and itself authenticated by the root.

VI. Simulation Setup

No public disaster dataset provides both exact per-pixel ground truth and the radiometric control needed to sweep noise, so a synthetic bi-temporal generator was written. Each 256×256 scene has a random fractal terrain incised by a meandering river, 40 rectangular buildings, an inundation extent obtained by thresholding elevation and keeping the component connected to the river, and damage assigned to 35% of buildings outside the flood. On average 22.3% of pixels are new flood water and 1.8% are damaged buildings, so damage is a minority class. Rubble is rendered with reduced reflectance and 45% multiplicative texture. The post-event scene receives a gain of 1.12, an offset of 0.015, a sub-pixel misregistration of (0.4, −0.3) pixels and additive Gaussian noise. Signal-to-noise ratio (SNR) is defined per band relative to the clean band variance, which makes the low-contrast green band the noise-limited one. Table II lists the configuration.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Scene size, bands	256 × 256 px (1024 × 1024 for timing); green, NIR
Scenes	20 seeds × 9 SNR levels (0–25 dB) = 180
Reflectance (G, NIR)	land 0.10, 0.32; water 0.07, 0.03; roof 0.24, 0.27; rubble 0.16, 0.19
Enhancement	$\rho = 0.7$, 3 iterations, 3 × 3 median
Damage scoring	5 × 5 window, min. footprint 20 px, score floor 0.02
Morphology	opening and closing, unit disc; min. area 25 px
Ledger	SHA-256; tile $T = 32$ px; anchor $K = 10$ unless swept
Tamper trials	500 per artefact or record case; 2000 per anchor setting
Software	Python 3, NumPy 2.5, SciPy 1.18, scikit-image 0.26
Hardware	2-core cloud Linux VM, single thread
Global seed	20261016

Baselines are the absolute near-infrared difference and the absolute log-ratio, each thresholded by Otsu [7], [8], evaluated on the binary damage mask. Accuracy is reported as F1 score, intersection over union (IoU) and Cohen's κ. The experiments were run in Python; MATLAB scripts implementing the same chain and ledger accompany the code but were not used to produce any number reported here.

VII. RESULTS AND DISCUSSION

A. Segmentation Quality

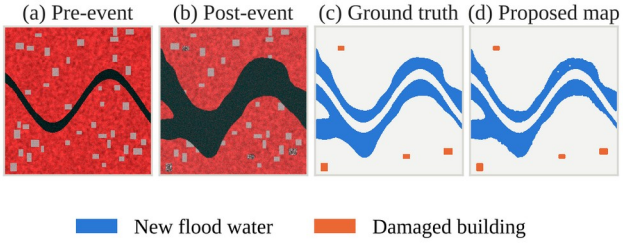


Fig. 2. Example scene at 12 dB SNR (false colour: NIR, green, green). The proposed map recovers the flood extent and the damaged structures outside it.

Fig. 2 shows a representative scene, and Fig. 3 the binary damage F1 score against SNR. From 10 dB upward the proposed chain is the most accurate method: at 15 dB it reaches F1 = 0.980, IoU = 0.961 and $\kappa = 0.974$, against 0.953, 0.910 and 0.939 for the difference baseline. The baselines saturate near 0.95 because a single change image cannot separate rubble from residual radiometric change. Per class, flood F1 is 0.980 and building-damage F1 0.976 at 15 dB, with building precision 0.998 and recall 0.956. Recall stays at about 0.956 at every SNR from 12 dB upward, which points to a structural miss, namely footprints whose eroded interior is empty or touches the flood buffer, rather than to noise.

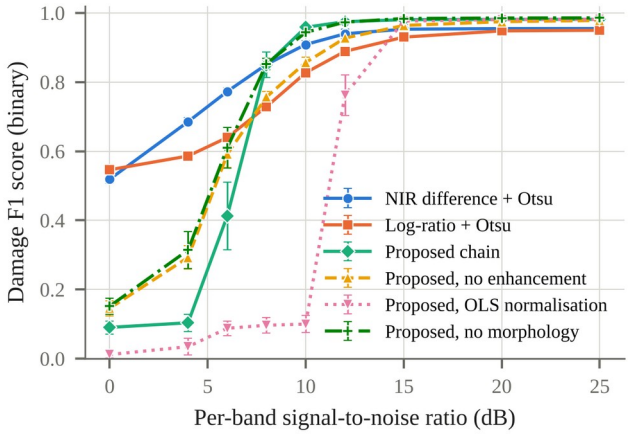


Fig. 3. Binary damage F1 score versus per-band SNR (mean ± one standard deviation over 20 scenes per level). Dashed and dotted curves are ablations of the proposed chain.

The chain fails below 8 dB. At 6 dB its F1 falls to 0.413 ± 0.098, while the difference baseline holds 0.773; at 8 dB the two are tied (0.850 against 0.851). The cause is physical: the NDWI depends on the green band, whose water-land contrast is about 0.02 in reflectance, so it is the first quantity to drown in noise, while the near-infrared contrast used by the baseline is about 0.28. An operational system should therefore record the scene SNR alongside the map and fall back to, or flag, a near-infrared rule in this regime. The result is retained because a damage map produced outside a method's operating range is exactly the kind of decision an audit trail should expose.

TABLE III
BINARY DAMAGE F1 SCORE OF BASELINES AND ABLATIONS

Method	6 dB	10 dB	15 dB	25 dB
NIR difference + Otsu	0.773	0.909	0.953	0.955
Log-ratio + Otsu	0.640	0.827	0.930	0.950
Proposed chain	0.413	0.958	0.980	0.981
without enhancement	0.591	0.856	0.964	0.978
with OLS normalisation	0.088	0.100	0.976	0.981
without morphology	0.610	0.944	0.984	0.986

Table III isolates each component. Enhancement helps from 10 dB upward (0.958 against 0.856 without it) but hurts at 6 dB (0.413 against 0.591), because PIF selection by residual becomes unreliable when noise is comparable to the change. OLS normalisation collapses the chain to 0.100 at 10 dB and only recovers at 15 dB, as Eq. (3) predicts. Morphological cleaning gives no benefit at or above 12 dB and costs about 0.005 in F1 at 25 dB by eroding thin flood margins; it is kept in the default configuration for its small gain at 10 dB (0.958 against 0.944), but this is a marginal choice.

B. Tamper Detection and Localisation

TABLE IV
TAMPER DETECTION RESULTS

Scenario	Detected	Localised
Map, 1 px relabelled	100%	tiles P/R 1.000/1.000
Map, 100 px relabelled	100%	tiles P/R 1.000/1.000
No tampering (false alarm)	0%	—
Source scene substituted	100%	ingest record
Record body edited	100%	±1 record, 100%
Record edited and rehashed	100%	±1 record, 100%
Record deleted / reordered	100% / 100%	±1 record, 100%
Suffix rewritten, no anchor	0.0%	—
Suffix rewritten, $K = 10$	96.8%	up to last anchor
Suffix rewritten, $K = 100$	80.7%	up to last anchor

Table IV lists the outcomes. Every artefact and record tampering case was detected, including a single relabelled pixel, and tile localisation had precision and recall of 1.000 for 1 to 100 altered pixels, as Proposition 1 requires. No untampered artefact was flagged. These results are by construction of a collision-resistant hash and confirm the implementation rather than discover a property.

The informative case is the storage adversary. Without anchoring, a rewritten suffix was never detected: the chain re-verified perfectly, so a ledger operated by the same agency that issues the maps adds no protection against that agency. Fig. 4(a) shows detection against the anchor interval. Monte Carlo rates of 96.8%, 88.7% and 60.1% for $K = 10$, 50 and 250 agree with Eq. (9), which gives 97.1%, 88.1% and 59.3%, within the binomial error of 2000 trials. Since anchoring a 32-byte digest is cheap, the design implication is to anchor at every issued map, which corresponds to $K = 1$ and full detection.

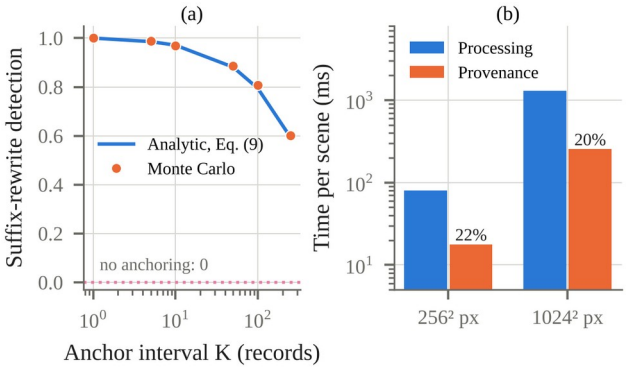


Fig. 4. (a) Detection probability of a storage-level suffix rewrite against the anchor interval K, Monte Carlo (2000 trials) and Eq. (9). (b) Processing and provenance time per scene; labels give provenance time as a share of processing time.

C. Re-execution Verification

Re-execution was tested by running the chain in single precision instead of double precision, a proxy for a different numerical platform, on 40 scenes at 10 and 20 dB. Table V shows that the floating-point NDWI product never reproduced its digest, raw or quantised to 10^{-6} or 10^{-4}, and reproduced only 55% of the time at a step of 10^{-2}, although the largest element difference was 1.4×10^{-6}. Eq. (10) with the measured mean error of 7.8×10^{-8} predicts 61% at 10^{-2}, in good agreement. The class maps, by contrast, were bit-identical in all 40 runs. The first design of this pipeline assumed that quantised digests of intermediates would serve re-execution; that assumption was wrong and is corrected here. Floating-point intermediates are committed for integrity, as produced, while bit-exact re-verification is applied to class maps, with a tolerance check such as IoU as a documented fallback.

TABLE V
DIGEST AGREEMENT AFTER RE-EXECUTION IN SINGLE PRECISION

Artefact digest	Agreement	Eq. (10)
NDWI, raw float	0%	—
NDWI, $q = 10^{-6}$	0%	0%
NDWI, $q = 10^{-4}$	0%	0%
NDWI, $q = 10^{-2}$	55%	61%
Class map D	100%	—

D. Computational Overhead

Fig. 4(b) compares processing and provenance time. For a 256 × 256 scene, processing took 80 ms and provenance 18 ms; for 1024 × 1024, 1308 ms and 256 ms, an overhead of about 20%. Raw SHA-256 throughput was 395 MB/s, so the cost lies in per-tile serialisation in an interpreted loop rather than in the hash itself, and a compiled implementation should reduce it substantially. A record append took 26 µs and a record averaged 506 bytes. Verifying 100 000 records took 642 ms. For a 1024 × 1024 map, 16-pixel tiles cost 56 ms and 128 kB of leaves, 32-pixel tiles 16 ms and 32 kB, and 128-pixel tiles 4 ms and 2 kB; $T = 32$ localises an edit to about 10 m on 0.3 m imagery at modest cost.

VIII. COMPARATIVE ANALYSIS

Table VI compares logging policies on the same assessment, including one analyst revision. Recording only the map digest recovers 1 of the 17 provenance elements ($A = 0.059$); adding source digests raises this to 0.176. Neither can show which parameters or code produced a map, so a silent re-run with a changed threshold is indistinguishable from the original. Stage-level recording reaches $A = 1.000$ with 7 records, and only the anchored variant resists the storage adversary. Model-level comparison with published systems is not meaningful because none reports these quantities for image products; Table VI therefore compares capabilities of policies implemented and measured here.

TABLE VI
LOGGING POLICIES COMPARED ON ONE ASSESSMENT

Policy	A	Map edit	Source swap	Param. or code change	Operator rewrite
Map digest only	0.059	Yes	No	No	No
Inputs + map digest	0.176	Yes	Yes	No	No
All stages, no anchor	1.000	Yes	Yes	Yes	No
Proposed (anchored)	1.000	Yes	Yes	Yes	Up to last anchor

IX. LIMITATIONS

All results come from synthetic two-band optical scenes. The generator lacks clouds, shadows, synthetic aperture radar speckle, terrain effects and the class imbalance of real events, so absolute F1 values should not be compared with results on xBD [12] or Sen1Floods11 [11]; validation on such data is the next step. The chain uses hand-set parameters and no learning, which keeps it reproducible but limits accuracy at low SNR, where it was outperformed by a simpler baseline. The ledger is a single-process model: replication, access control and key management were not implemented, records are not signed, so analyst identity is asserted rather than proven, and the witness was simulated.

The guarantees therefore start at commitment time; a compromised processing host can commit falsified products that will verify correctly. Timing figures come from an interpreted single-threaded implementation.

X. Conclusion

This paper has shown that the auditable part of a disaster damage map can be made precise. Committing every artefact through tile-level Merkle roots, binding each stage's code and parameters to its output, and anchoring the ledger head externally detected every local tampering case tested, localised it to the altered tile or record, and limited a storage-level rewrite to the records after the last anchor, as the derived exposure model predicts. The work also established two negative findings with design consequences: a hash chain without an external anchor gives no protection against its own operator, and quantised digests do not make floating-point intermediates reproducible, so re-verification should target class maps. The damage-mapping chain was accurate above 10 dB SNR and unreliable below 8 dB. Future work will validate the pipeline on Sentinel-1 and Sentinel-2 flood events and on xBD building damage, add analyst signatures and a public anchoring service, and port the commitment routine to compiled code.

Acknowledgment

The authors thank the Research and Development Division of Tekstein Solutions Private Limited for computing support. [Funding and sponsor details to be confirmed by the authors.]

References

[1] Copernicus Emergency Management Service, “Rapid mapping,” European Commission. [Online]. Available: https://emergency.copernicus.eu/

[2] International Charter Space and Major Disasters. [Online]. Available: https://disasterscharter.org/

[3] B. Tellman *et al.*, “Satellite imaging reveals increased proportion of population exposed to floods,” *Nature*, vol. 596, pp. 80–86, 2021.

[4] S. Haber and W. S. Stornetta, “How to time-stamp a digital document,” *J. Cryptology*, vol. 3, no. 2, pp. 99–111, 1991.

[5] R. C. Merkle, “A digital signature based on a conventional encryption function,” in *Proc. CRYPTO ’87*, LNCS 293, 1988, pp. 369–378.

[6] S. A. Crosby and D. S. Wallach, “Efficient data structures for tamper-evident logging,” in *Proc. 18th USENIX Security Symp.*, 2009, pp. 317–334.

[7] L. Bruzzone and D. F. Prieto, “Automatic analysis of the difference image for unsupervised change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[8] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[9] A. A. Nielsen, “The regularized iteratively reweighted MAD method for change detection in multi- and hyperspectral data,” *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 463–478, Feb. 2007.

[10] S. K. McFeeters, “The use of the normalized difference water index (NDWI) in the delineation of open water features,” *Int. J. Remote Sens.*, vol. 17, no. 7, pp. 1425–1432, 1996.

[11] D. Bonafilia, B. Tellman, T. Anderson, and E. Issenberg, “Sen1Floods11: A georeferenced dataset to train and test deep learning flood algorithms for Sentinel-1,” in *Proc. IEEE/CVF CVPR Workshops*, 2020, pp. 210–211.

[12] R. Gupta *et al.*, “Creating xBD: A dataset for assessing building damage from satellite imagery,” in *Proc. IEEE/CVF CVPR Workshops*, 2019, pp. 10–17.

[13] B. Laurie, A. Langley, and E. Kasper, “Certificate transparency,” IETF RFC 6962, Jun. 2013.

[14] E. Androulaki *et al.*, “Hyperledger Fabric: A distributed operating system for permissioned blockchains,” in *Proc. 13th EuroSys Conf.*, 2018, Art. no. 30.

[15] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, “ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability,” in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, 2017, pp. 468–477.

[16] L. Moreau and P. Missier, Eds., “PROV-DM: The PROV data model,” W3C Recommendation, Apr. 2013.

[17] J. R. Schott, C. Salvaggio, and W. J. Volchok, “Radiometric scene normalization using pseudoinvariant features,” *Remote Sens. Environ.*, vol. 26, no. 1, pp. 1–16, 1988.

[18] W. A. Fuller, *Measurement Error Models*. New York, NY, USA: Wiley, 1987.