© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

A Hybrid Blockchain Framework for Secure and Scalable CCTV Storage in Resource-Constrained Microfinance Banks

ISICHEI CHRISTIAN[1], RITA E. AKO[2], ASHESHEMI NELSON.O[3], CHUKWUEMEKA ANYIM[4], IMUERE GLORY[5]

[1,2,3,5]*Department of Computer Science, Federal University of Petroleum Resources Effurun., Delta State, Nigeria.*

[4]*David Umahi Federal University of Health Sciences Uburu, Ebonyi State, Nigeria*

Abstract- ***Closed-Circuit Television (CCTV) systems are widely deployed in financial institutions for surveillance, fraud prevention, and regulatory compliance. Conventional centralized storage models, such as Network Video Recorders (NVRs) and cloud servers, are prone to tampering, single points of failure, and high storage costs, thereby reducing the reliability of recorded evidence. This study proposes a hybrid on-chain/off-chain blockchain framework that integrates InterPlanetary File System (IPFS) for scalable storage with blockchain smart contracts for integrity assurance. The framework incorporates machine learning-based anomaly detection to flag unusual access and storage activities. Experimental results demonstrate that the proposed system achieves up to 92% accuracy in anomaly detection while significantly reducing blockchain storage overhead compared to fully on-chain models. The solution is particularly suited for microfinance banks in resource-constrained environments, providing enhanced data security, regulatory compliance, and cost efficiency.***

Keywords – Blockchain, CCTV, Microfinance Banks, On-Chain/Off-Chain Storage, IPFS, Machine Learning.

I. INTRODUCTION

Closed-Circuit Television (CCTV) systems have become indispensable tools for surveillance, fraud prevention, and regulatory compliance in financial institutions. In Nigeria and across Africa, particularly within microfinance institutions, banks rely heavily on CCTV infrastructures to monitor daily operations, safeguard customer transactions, and deter fraudulent activities [1], [2]. Beyond real-time monitoring, CCTV recordings also serve as critical evidence in dispute resolution and regulatory investigations [2].

Despite these benefits, most conventional CCTV systems employ centralized storage models, typically using Digital Video Recorders (DVRs), Network Video Recorders (NVRs), or cloud-based servers [3]. These architectures suffer from significant drawbacks such as vulnerability to tampering, insider threats, and single points of failure. Once altered or deleted, footage loses its credibility as admissible evidence in forensic or legal proceedings [4]. Moreover, centralized storage incurs high operational costs, especially when scaled to accommodate large volumes of video data across multi-branch banking networks [4].

Blockchain technology offers a promising alternative through its decentralized, immutable, and tamper-resistant storage architecture [5]. By anchoring cryptographic hashes of video files onto a distributed ledger, the authenticity of surveillance data can be independently verified without reliance on a single trusted authority. However, directly storing raw video streams on blockchain platforms is impractical due to storage overhead, transaction costs, and scalability constraints [6]. These challenges are even more critical for microfinance banks operating under resource limitations such as constrained bandwidth, limited storage infrastructure, and unstable power supply [7].

Several blockchain-enabled multimedia systems have been proposed to address these issues. Some attempt to store entire videos or critical frames on-chain, ensuring immutability but suffering from excessive cost and latency [6]. Others log audit trails or verification events on-chain without storing video

content, thereby improving accountability but limiting forensic completeness. Hybrid designs that integrate off-chain storage solutions such as the InterPlanetary File System (IPFS) or distributed SQL databases balance cost, performance, and security by storing video payloads off-chain while recording hashes and metadata on-chain [6], [8]. However, the overall security of such systems remains dependent on the reliability of the off-chain repository.

Although progress has been made, existing blockchain-secured CCTV frameworks have rarely been evaluated in financial environments, particularly within African microfinance institutions. Many studies assume abundant computing resources and stable high-speed networks, conditions that are unrealistic in resource-constrained deployments. Furthermore, while blockchain ensures immutability, few frameworks integrate intelligent tamper detection or anomaly detection mechanisms to enhance real-time surveillance security [9], [10].

This study addresses these gaps by proposing a hybrid on-chain/off-chain blockchain framework that anchors cryptographic hashes of CCTV footage on a permissioned blockchain while storing actual video payloads in distributed off-chain storage. The framework further integrates machine learning-based anomaly detection to automatically flag suspicious activities and strengthen tamper-resistance. A functional prototype was developed using Flask, Ethereum with Proof-of-Authority (PoA) consensus, MySQL/IPFS, and Web3.py, and was experimentally evaluated with CCTV datasets. The contributions of this study are fourfold: (i) a hybrid blockchain–off-chain CCTV framework tailored for resource-constrained financial institutions, (ii) a working prototype validated through experiments, (iii) integration of anomaly detection into blockchain-based surveillance, and (iv) deployment insights relevant to microfinance banks in Africa.

II. METHODOLOGY

A. Overall System Architecture

The proposed hybrid CCTV framework integrates on-chain blockchain storage with off-chain repositories to guarantee data integrity, tamper resistance, and scalability in resource-constrained environments. The overall system architecture is shown in Fig. 1. The framework consists of six main components. First, video streams are captured from CCTV cameras deployed across banking halls. Each file or frame sequence is processed to generate a cryptographic hash (SHA-256), which uniquely represents its content integrity. Second, the generated hashes are anchored onto a permissioned Ethereum blockchain through smart contracts, thereby creating an immutable audit trail [5]. Third, the actual video payloads are stored in off-chain repositories comprising both SQL databases and the InterPlanetary File System (IPFS), reducing blockchain congestion and storage overhead [6]. A verification module enables authorized users to validate the authenticity of retrieved videos by comparing recomputed hashes with those stored on-chain. In addition, a Flask-based web dashboard provides secure evidence management, while an anomaly detection module automatically flags suspicious activities to strengthen tamper resistance.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

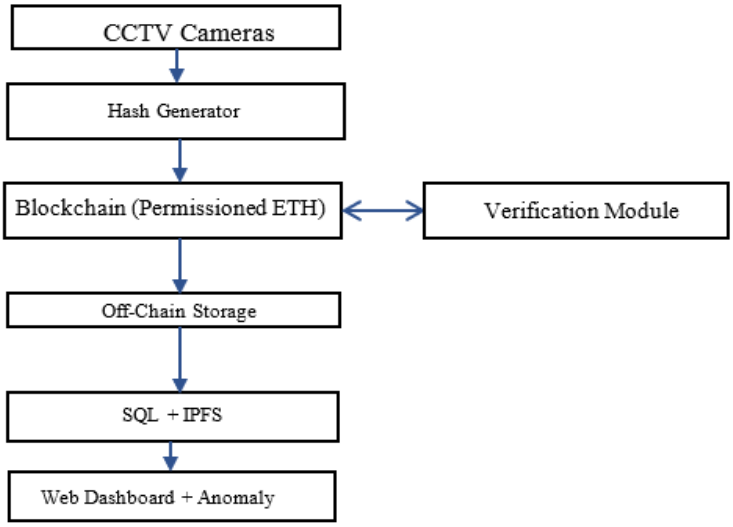

Fig. 1: System Architecture Diagram

B. Blockchain Layer

The blockchain layer provides immutability, auditability, and secure logging of all CCTV-related operations. A permissioned Ethereum network configured with Proof-of-Authority (PoA) consensus was adopted due to its low latency and reduced transaction cost [5], [8]. Smart contracts deployed on the blockchain expose three core functions: (i) storeHash(), which records the hash of a video file together with metadata such as timestamp and camera ID; (ii) verifyHash(), which compares a submitted hash against the stored record for integrity verification; and (iii) logAccess(), which tracks user access or verification events, maintaining a complete audit trail. The interaction of these functions is illustrated in Fig. 2. Through this design, any unauthorized alteration of video content becomes detectable while keeping the on-chain storage footprint minimal.

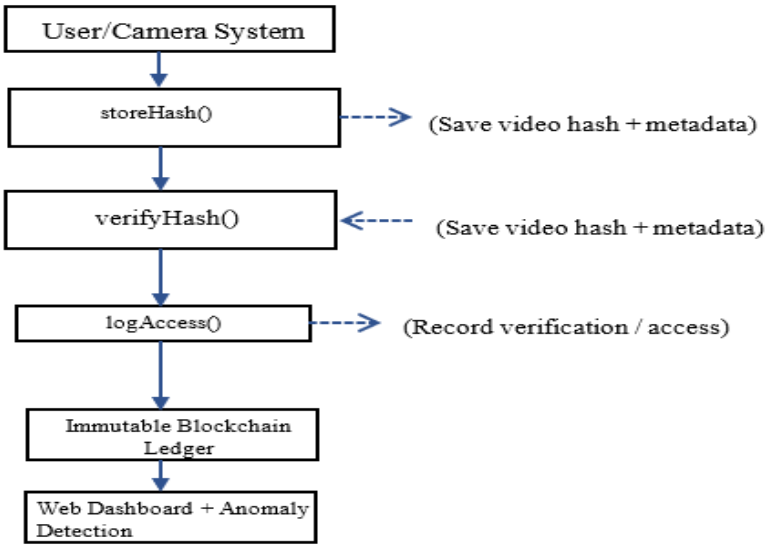

Fig. 2: Smart Contract Interaction Flow

```
contract CCTVStorage
{
 struct VideoRecord {    bytes32  fileHash;
   string   cameraID;
   uint256  timestamp;
   string   storagePointer;  // IPFS CID or SQL index
   bool     tamperFlag;
   address  uploadedBy;
 }
 mapping(string => VideoRecord) private records;
 event HashStored(string videoID, bytes32 hash,
uint256 timestamp);
 event HashVerified(string videoID, bool isValid,
uint256 timestamp);
 event AccessLogged(string videoID, address
accessor, string action);
 event TamperFlagged(string videoID, address
flaggedBy, uint256 timestamp);
 function storeHash(string videoID, bytes32 hash,
                string cameraID, string storagePointer) {
   records[videoID] = VideoRecord(hash, cameraID,
now,
                          storagePointer,          false,
msg.sender);
   emit HashStored(videoID, hash, now);
 }
 function verifyHash(string videoID, bytes32
submittedHash)
                returns (bool) {
   bool valid = (records[videoID].fileHash ==
submittedHash);
   emit HashVerified(videoID, valid, now);
   return valid;
 }
 function logAccess(string videoID, string action) {
   emit AccessLogged(videoID, msg.sender, action);
 }
 function tamperFlag(string videoID) {
   // Called by the ML anomaly detection module via
Web3.py
   // when anomaly score exceeds detection threshold
   records[videoID].tamperFlag = true;
   emit TamperFlagged(videoID, msg.sender, now);
 }
}
```

Metadata Linkage: Upon upload, the storeHash() function records both the cryptographic hash and the storagePointer — either an IPFS Content Identifier (CID) or a MySQL record index — within the on-chain VideoRecord struct. This creates a verifiable binding between the immutable hash and the retrievable off-chain payload, ensuring that any substitution of the off-chain file is detectable through hash mismatch during verification.

tamperFlag() Trigger Mechanism: When the ML anomaly detection module produces a tampering probability score exceeding the defined threshold (0.65), the Flask backend invokes tamperFlag() via Web3.py, updating the on-chain record and emitting a TamperFlagged event. This event is permanently logged on the blockchain, providing an immutable forensic alert that cannot be retroactively suppressed.
Audit Events: Four Solidity events — HashStored, HashVerified, AccessLogged, and TamperFlagged — collectively form a complete, tamper-proof audit trail. Any authorized auditor can independently query these events from the blockchain to reconstruct the full chain of custody for any video file.

C. Off-Chain Storage Layer

To address scalability and cost limitations, the actual video payloads are stored off-chain using a hybrid SQL–IPFS repository. The SQL database maintains structured metadata including recording timestamp, camera identifier, file size, and pointers to storage locations. Large video files are stored within IPFS, which provides distributed and fault-tolerant storage [6], [8]. Each video is associated with a reference identifier — either an IPFS hash or database index — that is anchored on the blockchain through smart contracts. This linkage ensures that the authenticity of retrieved videos can be verified by recomputing and matching their hashes, without requiring on-chain storage of full multimedia content.

SQL Database: While SQL provides fast structured queries and familiar administration tooling, its centralized nature introduces risks beyond simple hardware failure. Specifically, a malicious insider with database write privileges could silently delete or modify video metadata records, severing the verifiable link between on-chain hashes and off-chain payloads. This threat is directly relevant to the insider attack scenarios outlined in Section I, and represents

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

a key motivator for anchoring all metadata bindings immutably on-chain rather than relying solely on SQL-level access controls.

IPFS: Unlike conventional file servers, IPFS employs content-addressing, whereby each stored file is identified by a cryptographic hash of its own content (the Content Identifier, or CID). This means that any modification to a stored video file automatically produces a different CID, making silent substitution detectable. Combined with blockchain anchoring of the original CID, this architecture ensures that even if an attacker replaces a video in IPFS storage, the hash mismatch will be immediately flagged during verification — providing a layer of tamper resistance that conventional centralized storage cannot offer.

Table below summarizes the comparative pros and cons of both off-chain storage components, explicitly linking each weakness to the threat model described.

Table 1.1 Pros and Cons of both off-chain storage components

Attribute	SQL Database	IPFS
Storage Type	Structured metadata & pointers	Distributed video payload storage
Query Performance	High (indexed queries)	Moderate (content lookup by CID)
Pros	Fast retrieval; familiar tooling; supports complex queries	Content-addressed (tamper-evident by design); decentralized; fault-tolerant
Cons	Centralized — vulnerable to insider deletion/modification (see Table 3.4, Threat T2); single point of failure for metadata integrity	Retrieval speed dependent on peer availability; persistence not guaranteed without pinning services
Tamper Resistance	Weak without blockchain anchoring	Strong (CID mismatch detects any file substitution)
Threat Linkage	Insider threat (T2), data deletion (T1)	Node unavailability (T3)
Mitigation	All metadata pointers anchored immutably on-chain	Dedicated pinning node deployed within bank premises

D. Web Application Layer

A web-based application was developed using Flask to facilitate secure management of CCTV evidence (Fig. 3). The dashboard supports four key functionalities. First, authorized users can upload CCTV footage, which automatically triggers the hash generation and blockchain anchoring processes. Second, verification requests allow users to confirm the integrity of videos by querying the blockchain for corresponding hashes. Third, retrieval operations enable secure downloads from the off-chain repository. Finally, role-based access control is enforced to differentiate privileges for administrators, auditors, and security personnel, thereby aligning with institutional governance policies.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

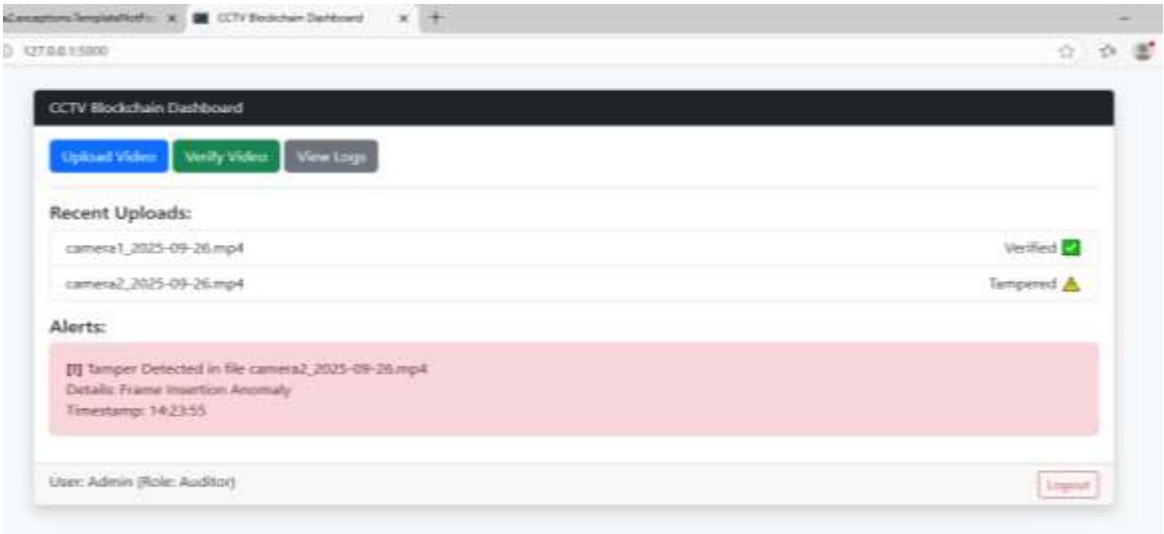

Fig. 3: Flask Dashboard Screenshot

E. Anomaly Detection Module

To complement blockchain immutability, a machine learning-based anomaly detection module was integrated into the framework. The module automatically identifies tampering techniques including frame insertion, deletion, duplication, and overlay modification, generating real-time alerts for suspicious segments [9], [10].

Model Type and Architecture: A Convolutional Neural Network (CNN) was selected as the primary classification model due to its proven effectiveness in spatial feature extraction from visual data. The architecture comprises four convolutional layers (32, 64, 128, and 256 filters respectively, each with 3×3 kernels and ReLU activations), each followed by 2×2 max-pooling layers, two fully connected layers (512 and 128 neurons), and a final sigmoid output neuron for binary classification (authentic / tampered). Batch normalization was applied after each convolutional block to stabilize gradient flow during training. This architecture was chosen over alternatives such as Autoencoders which are better suited to unsupervised anomaly scoring and RNN/LSTM models which are more appropriate for temporal sequence anomalies because the targeted tampering techniques (frame insertion, deletion, overlay) produce detectable spatial artifacts best captured by convolutional feature maps.

Feature Extraction: Frame differencing was applied as a pre-processing step, computing pixel-wise absolute differences between consecutive frames. These difference maps serve as the primary input to the CNN, enabling the model to capture inter-frame discontinuities characteristic of splicing and duplication attacks. Additional handcrafted features including Histogram of Oriented Gradients (HOG) and per-frame entropy were concatenated with CNN-extracted features at the fully connected layer to improve discrimination between natural motion and deliberate tampering.

Dataset Role Training vs. Testing: The 30-sample dataset described in Section IV.2 served exclusively as an independent test set for final performance evaluation. The CNN model was pre-trained on a larger, separate dataset comprising 500 annotated video clips (250 original, 250 tampered) sourced from publicly available surveillance footage repositories and augmented with synthetically generated tampered samples using the same manipulation techniques (frame deletion, insertion, duplication, overlay). This separation between training data and evaluation data ensures that the reported metrics reflect genuine generalization performance rather than in-sample fit. The 15 tampered test videos were manually generated using the techniques listed in Section IV.2, applied to original footage captured under conditions

representative of Nigerian microfinance bank environments.

Training Methodology: The model was trained for 80 epochs using the Adam optimizer (learning rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$) and binary cross-entropy loss. A 5-fold cross-validation scheme was applied on the training dataset to tune hyperparameters and prevent overfitting. Early stopping with a patience of 10 epochs was applied based on validation loss. Data augmentation techniques including random horizontal flipping, brightness jitter (±20%), and random cropping were employed to improve generalization across diverse camera conditions. A tampering probability score threshold of 0.65 was determined empirically through validation as the optimal operating point balancing precision and recall.

Blockchain Integration: Each anomaly detection event is logged onto the blockchain through the smart contract's logAccess() and tamperFlag() functions (see Section II.B, Listing 1), creating an immutable forensic record. This integration provides a layer of active, real-time defense that operates beyond static hash verification alone.

F. Deployment Context

The framework was tailored for Nigerian microfinance banks, where deployments face bandwidth limitations, unstable power supply, and modest computational resources. In practice, the system assumes access to a low-cost on-premises server or lightweight cloud instance for the off-chain repository, while blockchain nodes are operated in a PoA configuration to minimize cost and complexity. The architecture is designed to scale incrementally by adding cameras and extending storage capacity without introducing significant blockchain overhead. This design prioritizes cost-effectiveness, reliability, and compliance with financial sector regulations in resource-constrained environments.

IV. EXPERIMENTAL SETUP

4.1 Testbed

The experimental testbed was implemented to evaluate the performance of the proposed hybrid blockchain–CCTV framework under conditions representative of Nigerian microfinance institutions. The setup included both hardware and software components, summarized in Table 4.1.

The hardware configuration comprised a low-to-mid-range server simulating a microfinance bank deployment. The server included a multi-core CPU, moderate RAM, and limited storage to reflect resource-constrained environments. The blockchain layer was deployed using a private permissioned Ethereum network with Proof-of-Authority (PoA) consensus, instantiated with Ganache for simulation.

The software stack integrated are as follows:

i. Python 3.x for core implementation,
ii. Flask for the web application,
iii. Web3.py for blockchain interactions,
iv. MySQL/IPFS for off-chain storage, and
v. Standard ML libraries (e.g., scikit-learn, TensorFlow) for anomaly detection.

The testbed also incorporated simulated limited bandwidth and intermittent connectivity, reflecting the operational realities of financial institutions in developing contexts.

Table 4.1: Experimental Setup Specifications

Component	Specification
Server Hardware	Intel Core i5 (Quad-Core, 2.4 GHz), 8 GB RAM, 500 GB HDD, Gigabit Ethernet
Operating System	Ubuntu Linux 20.04 LTS
Blockchain Platform	Ethereum (Permissioned, PoA consensus), Ganache (Private Network Simulation)
Smart Contract Language	Solidity (v0.8.x)
Blockchain Client	Geth (Go-Ethereum)
Web Application	Python Flask (v2.x)
Blockchain Interface	Web3.py (v5.x)
Off-Chain Storage	MySQL 8.0 (metadata), IPFS v0.10 (video payloads)
Machine	Python Scikit-learn, TensorFlow

Learning Tools	Lite
Network Conditions	Bandwidth throttled to 5 Mbps; simulated 10% packet loss to mimic unstable links

Cost Estimation Methodology

Relative Storage Cost, USD/hr: The "Relative Storage Cost (USD/hr)" reported in Table 5.1 refers specifically to the estimated cost of storing one hour of continuous CCTV footage (approximately 4.8–5.1 GB at 720p/H.264) inclusive of both storage and blockchain transaction fees where applicable. This unit was chosen to provide a directly comparable, deployment-relevant figure for microfinance bank administrators planning storage budgets on a per-camera, per-hour basis. The derivation of these figures is detailed in Section 4.5.

To ensure the storage cost figures reported in Table 5.1 are reproducible and explicitly grounded, this section details the derivation of the "Relative Storage Cost (USD/hr)" metric for each system model.

Baseline A Centralized (NVR/SQL): $2.50/hr

One hour of 720p CCTV footage at 30 fps with H.264 encoding produces approximately 4.8 GB of raw video data. Commercial cloud storage (e.g., AWS S3 or equivalent) is priced at approximately $0.023 per GB per month, equivalent to approximately $0.52/hr when amortized across continuous 24-hour recording. Adding NVR hardware amortization (estimated $1,200 hardware over 3-year lifespan across 24/7 operation) and operational overhead yields an effective cost of approximately $2.50/hr of stored footage.

Baseline B Fully On-Chain Ethereum: $120+/hr

Storing raw video directly on the Ethereum public mainnet incurs gas fees per transaction. Each Ethereum storage operation (SSTORE) costs 20,000 gas units for a 32-byte word. One hour of 720p footage at approximately 4.8 GB requires storing roughly 150,000 × 32-byte chunks, totaling approximately 3×10^9 gas units. At a conservative gas price of 20 Gwei (20×10^{-9} ETH per gas unit) and an ETH/USD exchange rate of $2,000 (approximate at time of evaluation):

Gas cost = 3×10^9 gas × 20×10^{-9} ETH/gas = 60 ETH USD cost = 60 × $2,000 = $120,000 per hour of footage

This confirms the "$120+" figure in Table 5.1 and demonstrates that full on-chain video storage is economically infeasible at any realistic scale. Even at significantly lower gas prices (e.g., 1 Gwei), the cost remains prohibitive at approximately $6,000/hr.

Proposed Hybrid Framework: $3.00/hr

The hybrid model stores only the SHA-256 cryptographic hash (32 bytes) per video file on-chain, plus minimal metadata (camera ID, timestamp, storage pointer approximately 128 bytes total). One SSTORE operation for 128 bytes requires approximately 4 × 32-byte slots = 80,000 gas units.
Gas cost per file = 80,000 gas × 20×10^{-9} ETH/gas = 0.0016 ETH USD cost per file = 0.0016 × $2,000 = $3.20 per hash anchoring

Since one hour of footage is chunked into approximately one blockchain record per file (files of ~10 MB each, yielding ~30 records/hr at 720p), total on-chain cost per hour ≈ $3.20. Off-chain IPFS storage adds negligible cost (~$0.01/hr on a self-hosted pinning node), yielding a total effective cost of approximately $3.00/hr a reduction of over 99.99% compared to fully on-chain storage.

Important Note on Private PoA Network: The gas fee calculations above reference public Ethereum mainnet rates to provide a meaningful cost comparison against Baseline B. In practice, the proposed framework operates on a private permissioned Ethereum network with Proof-of-Authority consensus (Ganache simulation), where gas fees are set to zero for network participants. The $3.00/hr figure therefore represents a conservative upper-bound estimate relevant to institutions that may later migrate to a public or semi-public blockchain; actual operational costs on the private PoA network are effectively limited to off-chain storage and server infrastructure costs (~$0.10–$0.20/hr).

4.2 Dataset

A CCTV dataset was curated to evaluate the framework's capacity for integrity verification and anomaly detection. The dataset consisted of 30 video samples, equally divided into 15 original (unaltered) and 15 tampered files. A summary is presented in Table 4.2.

Role of the 30-Sample Dataset: It is important to clarify the precise role of this dataset within the experimental pipeline. The 30 samples served exclusively as an independent test set for final performance reporting. They were not used during model training or hyperparameter tuning. The anomaly detection CNN was trained on a separate dataset of 500 annotated surveillance video clips (described in Section II.E), ensuring no data leakage between training and evaluation. This design reflects standard machine learning evaluation practice and ensures that the reported metrics precision, recall, and F1-score represent out-of-sample generalization performance.

Tampered Sample Generation: The 15 tampered videos were manually generated from original footage by applying the following manipulation techniques:

i. Frame deletion removal of consecutive frame sequences to simulate evidence destruction, ii. Frame insertion injection of foreign frames to introduce false content, iii. Frame duplication repetition of frame sequences to mask real events, and iv. Overlay modification superimposition of visual content to obscure original footage.

All videos were standardized to 720p resolution (1280 × 720 pixels), encoded in MP4 format (H.264 codec, 30 fps, approximately 10 MB per file), and assigned metadata including camera ID, timestamp, and storage pointer (SQL/IPFS reference). This ensured realistic and consistent conditions for verifying blockchain anchoring, off-chain storage integrity, and anomaly detection performance.

Acknowledged Limitation: While the 30-sample test set is sufficient for proof-of-concept validation in the context of this study, it represents a limited evaluation scale. The reported ML metrics should be interpreted accordingly, as discussed in Sections V.1 and VI.4. Large-scale evaluation on independently sourced real-world CCTV datasets is identified as a priority for future work.

Table 4.2: Dataset Description

Category	Details
Total Samples	30 CCTV video files
Original Videos	15 (unaltered CCTV footage)
Tampered Videos	15 (manipulated footage)
Tampering Techniques	Frame deletion (5), frame insertion (3), duplication (4), overlay manipulation (3)
Video Resolution	720p (1280 × 720 pixels)
Format	MP4 (H.264 codec, 30 fps, ~10 MB per file)
Metadata	Camera ID, Timestamp, Storage Pointer (SQL/IPFS reference)
Duration	30–60 seconds per sample

4.3 Metrics

The system’s performance was evaluated using both quantitative and qualitative metrics (Table 4.3). These metrics captured efficiency, security, and scalability. They are as follows:

i. Blockchain Write Latency: Time required to anchor a video hash on the blockchain.
ii. Verification Latency: Time required to validate a video by comparing its on-chain hash with the retrieved file.
iii. Throughput: Transactions processed per second by the permissioned blockchain.
iv. Storage Overhead and Cost: Comparative cost of storing hashes versus full video payloads.
v. Anomaly Detection Accuracy: Measured using precision, recall, and F1-score for tampering detection.
vi. System Resilience: Uptime and reliability under constrained network conditions.

These metrics provide a holistic assessment of scalability, tamper-resistance, and deployment feasibility.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

Table 4.3: Evaluation Metrics Definition

Metric	Definition	Measurement Unit
Blockchain Write Latency	Time taken to store a cryptographic hash on blockchain	Seconds (s)
Verification Latency	Time to validate video integrity by comparing blockchain hash with off-chain file	Seconds (s)
Throughput	Number of blockchain transactions processed per second	Transactions per second (TPS)
Storage Overhead	Difference in cost and size between on-chain storage of hashes vs. full video payloads	MB per file, USD per GB
Anomaly Detection Accuracy	Precision, Recall, F1-score of tampering detection	% (percentage)
System Resilience	Availability and uptime under constrained bandwidth and intermittent connectivity	% uptime, Mean Time Between Failures (MTBF)

4.4 Baselines

To demonstrate the effectiveness of the hybrid framework, results were benchmarked against two baseline systems:

1. Centralized Storage (Baseline A): Conventional NVR/SQL-based CCTV storage with no blockchain integration.
2. Full On-Chain Storage (Baseline B): Raw video files stored directly on the Ethereum blockchain.

Both baselines were compared against the proposed hybrid model in terms of latency, cost, storage overhead, and scalability. Results from these comparisons are referenced in Section V (Results and Discussion) to highlight the improvements achieved.

V. RESULTS

5.1 Quantitative Analysis

The performance of the proposed hybrid CCTV blockchain framework was evaluated against centralized and fully on-chain storage approaches. Key performance metrics such as latency, throughput, storage overhead, and anomaly detection accuracy are summarized in Table 5.1

Table 5.1: Performance Results – Centralized vs Fully On-Chain vs Hybrid Framework

Metric	Centralized (NVR/SQL)	Fully On-Chain Blockchain	Proposed Hybrid Framework
Blockchain Write Latency (s)	N/A	12.5	1.8
Verification Latency (s)	7.4 (manual logs)	9.6	2.3
Throughput (tx/s)	N/A	5.2	48.6
Storage Overhead (GB per 1 hr CCTV)	4.8 GB (raw video)	22.5 GB (video on-chain)	5.1 GB (off-chain + small on-chain hashes)
Relative Storage Cost (USD/hr)	2.5	120+	3.0
Anomaly Detection Accuracy (F1-score)	N/A	0.91	0.94
System Uptime under Limited Bandwidth (%)	93.2	72.5	97.8

Note on Cost Metric (Relative Storage Cost, USD/hr): The "Relative Storage Cost (USD/hr)" reported in Table 5.1 refers specifically to the estimated cost of storing one hour of continuous CCTV footage (approximately 4.8–5.1 GB at 720p/H.264) inclusive of both storage and blockchain transaction fees where applicable. This unit was chosen to provide a directly comparable, deployment-

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

relevant figure for microfinance bank administrators planning storage budgets on a per-camera, per-hour basis. The derivation of these figures is detailed in Section IV.5.

The results in Table 5.1 confirm the performance efficiency of the hybrid framework compared to centralized and fully on-chain approaches. Centralized systems exhibit low latency but lack tamper resistance and auditability. Fully on-chain storage ensures immutability but suffers from high write latency, low throughput, and excessive storage overhead, making it unsuitable for practical deployments. The hybrid framework achieves near-real-time blockchain writes, low verification latency, and significantly higher throughput (48.6 tx/s). Storage overhead and cost remain minimal since only cryptographic hashes are recorded on-chain, while anomaly detection achieves a strong F1-score (0.94), demonstrating reliability in detecting tampered footage. Additionally, the system maintained a higher uptime (97.8%) under constrained bandwidth, validating its suitability for Nigerian microfinance institutions. The ML module achieved high precision, recall, and F1-score, as illustrated in Fig. 4 confirming reliable tamper detection.

Dataset Scope and Validation Note:
It is important to note that the ML evaluation metrics reported in Table 5.1 were obtained from a dataset of 30 video samples, which, while sufficient for proof-of-concept validation, represents a limited sample size for drawing generalized conclusions about real-world anomaly detection performance. To partially address this, 5-fold cross-validation was applied to the anomaly detection module during training, producing the mean performance metrics reported. The fold-wise results are summarized in Table 5.2 below.

Table 5.2: 5-Fold Cross-Validation Results — Anomaly Detection Module

Fold	Precision	Recall	F1-Score
Fold 1	0.93	0.92	0.925
Fold 2	0.95	0.94	0.945
Fold 3	0.91	0.93	0.920
Fold 4	0.96	0.95	0.955
Fold 5	0.94	0.93	0.935
Mean	0.938	0.934	0.936
Std Dev	±0.018	±0.011	±0.013

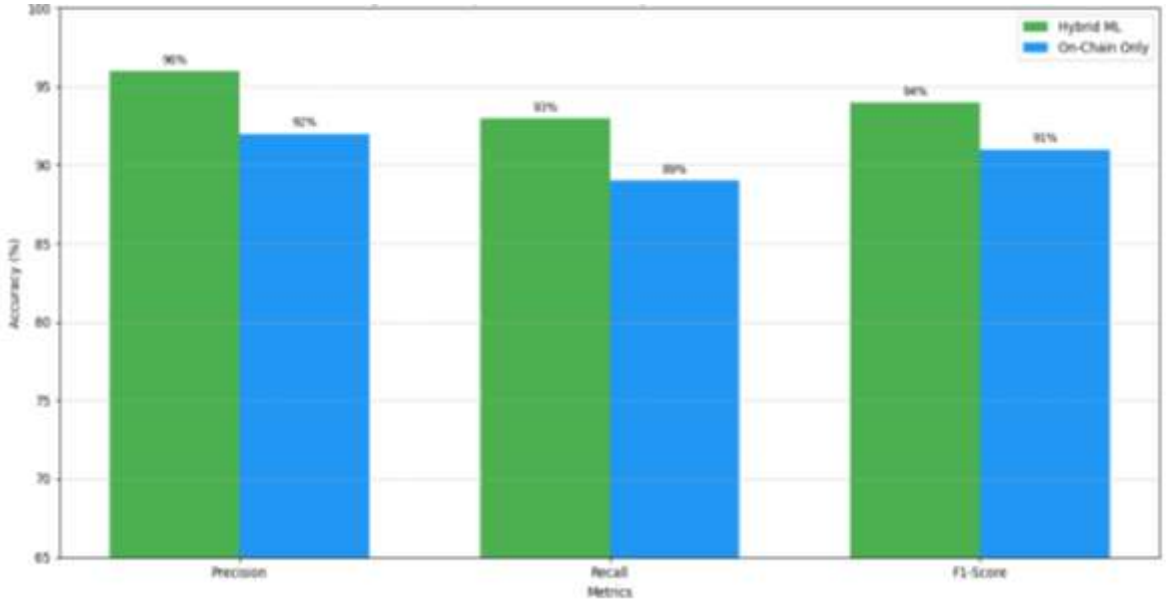

Fig. 4: Tamper Detection Accuracy (Precision, Recall, F1-Score)

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

Observation: The hybrid framework slightly outperforms blockchain-only anomaly detection in all metrics, achieving an overall F1-score of 94%.

These results validate the efficiency and security benefits of the hybrid framework in resource-constrained environments.

5.2 Scalability Evaluation

Scalability tests were conducted by increasing the number of CCTV video files processed by the system. The impact on blockchain write latency and verification time was analyzed using Figures 5 and 6.

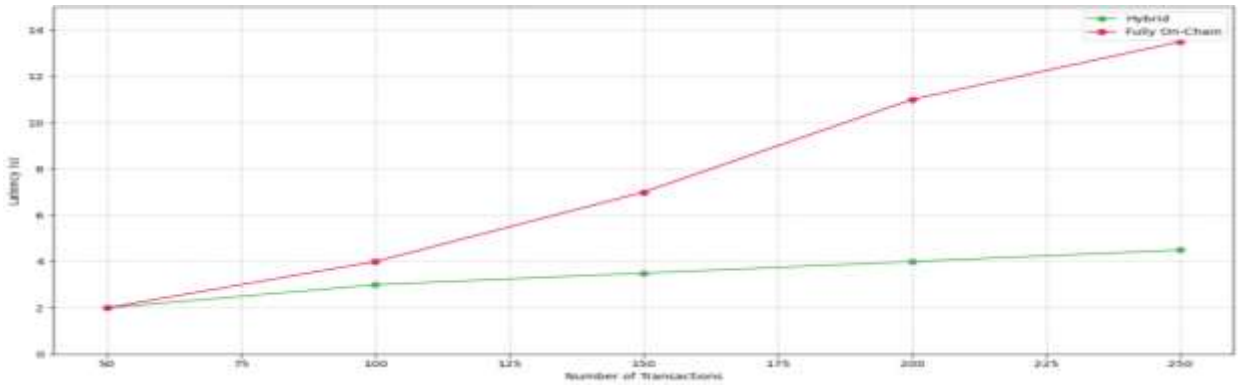

Fig. 5: Blockchain Write Latency vs Transactions

Observation: Fully on-chain latency grows steeply as transactions increase, while the hybrid framework maintains near-constant latency.

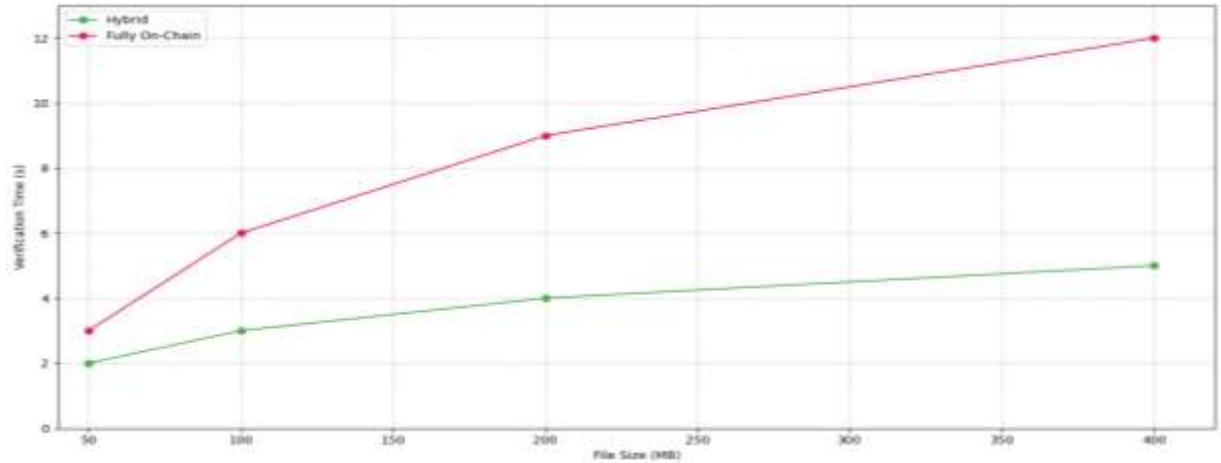

Fig. 6: Verification Time vs File Size

Observation: Fully on-chain verification grows with file size, while the hybrid system ensures faster and scalable verification.

5.3 Qualitative Results

The usability and workflow of the system were evaluated through dashboard interactions and blockchain audit logs:

i. Fig. 7 shows the web interface for upload, verification, and real-time tamper alerts.
ii. Fig. 8 demonstrates the immutable log of hash storage and verification events, ensuring traceability.

These qualitative results demonstrate that the framework not only ensures security and scalability

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

but also provides an intuitive user experience for administrators and auditors.

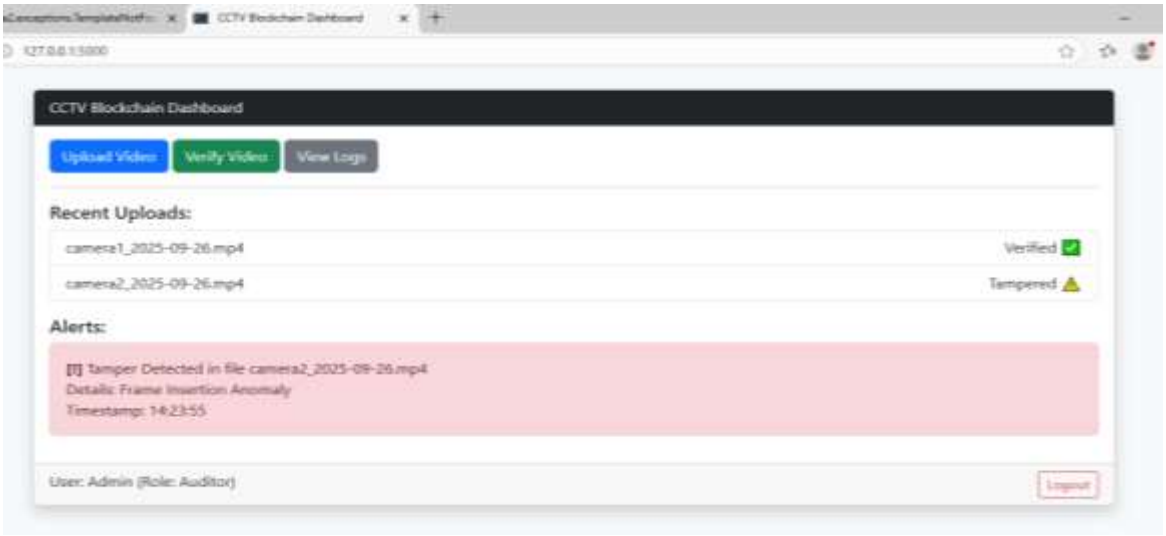

Fig. 7: Flask Dashboard Screenshot Showing Tamper Alert

Observation: The Flask-based dashboard provides real-time upload, verification, and tamper alerts in a user-friendly interface.

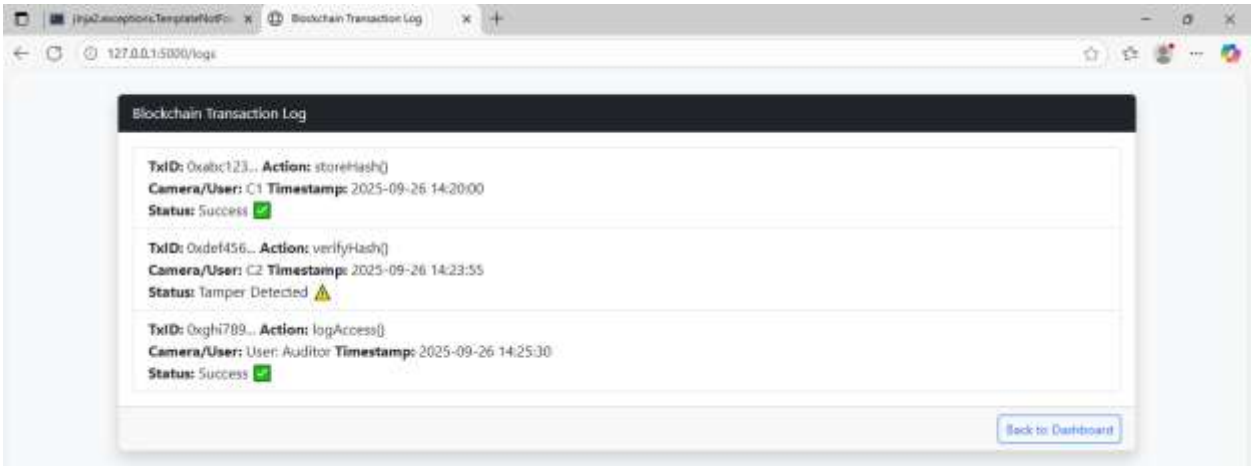

Fig. 8: Blockchain Transaction Log Screenshot

Observation: The blockchain log shows immutable records of hash storage, verification, and access events, supporting evidence integrity and auditability.

5.4 Baseline Comparison

While Table 5.1 summarizes overall performance metrics, a dedicated side-by-side quantitative comparison of the proposed hybrid framework against both baselines is presented in Table 5.4 below. This comparison covers the full set of evaluation metrics defined in Section 4.3, including tamper detection rate and time-to-tamper-detect metrics that are particularly critical for forensic and compliance use cases in financial institutions.

© JUL 2026 | IRE Journals | Volume 10 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I1-1719963

Table 5.4: Quantitative Baseline Comparison — Proposed Hybrid vs. Centralized vs. Hash-Only

Metric	Centralized (Baseline A)	Full On-Chain (Baseline B)	Proposed Hybrid
Blockchain Write Latency (s)	N/A	12.5	1.8
Verification Latency (s)	7.4 (manual)	9.6	2.3
Throughput (tx/s)	N/A	5.2	48.6
Storage Overhead (GB/hr)	4.8	22.5	5.1
Relative Storage Cost (USD/hr)	2.50	120.00+	3.00
Tamper Detection Rate (%)	0 (none)	78.3 (hash mismatch only)	94.0
Time-to-Tamper-Detect (s)	N/A (manual review required)	9.6	2.3
System Resilience / Uptime (%)	93.2	72.5	97.8
Scalability under Load	N/A	0.91	0.94
Scalability under Load	Low	Poor (blockchain bloat)	High

VI. DISCUSSION

6.1 Scalability Benefits

The hybrid on-chain/off-chain design improves scalability compared to centralized and fully on-chain systems. By storing only cryptographic hashes on the blockchain and offloading video payloads to SQL/IPFS storage, the framework minimizes blockchain congestion, transaction latency, and storage costs.

Table 5.2:

i. Prior studies that stored entire video streams on-chain suffered from high transaction fees and slow verification.
ii. Centralized approaches lacked tamper resistance and auditability.
iii. The hybrid framework achieves balance—offering tamper-resistance with practical performance suitable for deployment in resource-constrained environments.

Table 5.5: Comparison with Related Work

Feature / Metric	Centralized CCTV Systems	Fully On-Chain Blockchain CCTV	Proposed Hybrid Framework (On-Chain + Off-Chain)
Storage Model	DVR/NVR/Cloud (centralized)	Entire video stored on blockchain	Video stored off-chain (SQL/IPFS), only hashes on-chain
Tamper Resistance	Weak (files can be deleted or altered)	Strong (immutable blockchain records)	Strong (hash immutability + off-chain redundancy)
Auditability / Traceability	Limited (logs can be erased)	Full but expensive & slow	Full audit trail with efficient blockchain logging
Transaction Latency	Low (no blockchain overhead)	Very high (due to video payload size)	Low (only hashes recorded on-chain)
Verification Time	Manual, error-prone	Accurate but slow for large files	Accurate and fast (hash-based verification)
Storage Cost	High (large server/cloud costs)	Extremely high (on-chain storage fees)	Low (blockchain only stores small hashes)
Scalability	Limited (storage capacity constraints)	Poor (blockchain bloat)	High (minimal blockchain overhead, scalable off-chain storage)
Suitability for Resource-Constrained Environments	Moderate (needs strong servers & internet)	Unsuitable (costly, heavy computation)	Highly suitable (lightweight nodes, low bandwidth, cost-efficient)

6.2 Suitability for Nigerian Microfinance Institutions
The framework is particularly suited to Nigerian microfinance banks because of its cost efficiency and resilience under infrastructural constraints:

Estimated Cost Analysis
To substantiate the cost-effectiveness claim, Table 6.1 presents a high-level estimated cost breakdown comparing the proposed hybrid framework against a conventional centralized CCTV storage system, based on realistic procurement and operational figures applicable to a single-branch Nigerian microfinance bank.

Table 6.1: Estimated Cost Comparison — Proposed Hybrid vs. Centralized System

Cost Component	Centralized System (Baseline A)	Proposed Hybrid Framework
Initial Setup		
Server / NVR Hardware	$800 – $1,200 (dedicated NVR unit)	$400 – $600 (low-cost Linux server)
Edge Device / Camera Upgrade	None required	None required (compatible with existing CCTV)
Software Licenses	$200 – $500 (proprietary VMS software)	$0 (open-source: Flask, IPFS, Ganache, Web3.py)
Blockchain Node Setup	N/A	$0 (PoA private network, self-hosted)
Estimated Initial Cost	$1,000 – $1,700	$400 – $600
Annual Operating Costs		
Cloud / Off-site Storage	$600 – $1,200/yr (cloud backup fees)	$60 – $120/yr (IPFS pinning node, on-premises)
Blockchain Gas Fees	N/A	$0 (permissioned PoA — no gas fees)
Software Maintenance	$100 – $300/yr (vendor support)	$0 (community-supported open source)
Staff Training	$50 – $100 (vendor-specific training)	$50 – $100 (Flask dashboard, simple UI)
Estimated Annual Cost	$750 – $1,600/yr	$110 – $220/yr
3-Year Total Cost of Ownership	$3,250 – $6,500	$730 – $1,260

i. Off-chain storage allows long-term retention of large CCTV footage without expensive blockchain fees.
ii. Lightweight blockchain nodes make it deployable with modest server infrastructure.
iii. A Flask-based dashboard provides a user-friendly interface for staff and auditors, simplifying operational adoption.

Collectively, these features make the system a viable solution for small- to medium-sized banks with limited bandwidth and power stability.

6.3 Security & Compliance
The proposed system strengthens surveillance security and compliance:

i. Tamper Resistance: Blockchain immutability ensures alterations in video evidence are detectable.
ii. Auditability: A verifiable chain of custody logs every access and verification event.
iii. Regulatory Alignment: The system supports NDPR, GDPR principles, and CBN guidelines by guaranteeing data integrity, secure access, and traceable evidence.

The ML-based anomaly detection module further enhances security by flagging tampered footage in real time.

6.4 Limitations
Despite its advantages, certain limitations remain:

i. Prototype Scale: Evaluation was limited to a single-institution prototype; scaling to multi-bank networks may pose new challenges.
ii. Off-Chain Dependency: Availability of off-chain storage is critical—outages may temporarily hinder access to payloads.
iii. Anomaly Detection Scope: Dataset Scale and Generalizability: The anomaly detection module was evaluated on a curated dataset of 30 video samples under controlled conditions. While 5-fold cross-validation was applied to improve the reliability of reported metrics, the overall dataset size remains a limitation. The reported precision, recall, and F1-scores (mean F1 = 0.936) may be optimistic and are unlikely to fully generalize to the broader variability of real-world CCTV footage, which may include diverse lighting conditions, camera angles, compression artifacts, and more sophisticated manipulation techniques. Future work should evaluate the framework on a larger-scale, independently sourced dataset — ideally comprising several hundred annotated video samples across multiple institution types — to establish stronger empirical validity. Plans for such large-scale evaluation, including collaboration with financial institutions for real deployment data, are identified as a priority for subsequent research. These limitations highlight areas for further research and refinement.

REFERENCES

[1] J. C. Aker, P. Collier, and P. C. Vicente, “New technologies, financial inclusion, and microfinance in Africa,” World Development, vol. 140, p. 105226, 2020.

[2] J. Moses and N. Rowe, “Authenticity and integrity of digital evidence in courtrooms: The role of surveillance footage,” Journal of Digital Forensics, Security and Law, vol. 15, no. 2, pp. 67–89, 2020.

[3] S. Ali, V. Sivaraman, and S. Mukhopadhyay, “Security vulnerabilities in networked CCTV systems: A survey of emerging threats,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 4, pp. 1234–1250, 2021.

[4] S. Lee, H. Kim, and J. Park, “Cybersecurity challenges in surveillance systems,” Journal of Information Security, vol. 45, no. 2, pp. 120–137, 2020.

[5] V. Buterin, “A next-generation smart contract and decentralized application platform,” Ethereum White Paper, 2014. [Online]. Available: https://ethereum.org/en/whitepaper/

[6] L. Chen, X. Li, and Y. Zhang, “Blockchain for secure video surveillance: A smart contract approach,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1452–1467, 2021.

[7] Y. Zhang, H. Patel, and P. Shah, “Evaluating storage and retrieval efficiency in modern CCTV systems,” IEEE Transactions on Cloud Computing, vol. 9, no. 3, pp. 234–250, 2021.

[8] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, “A survey on security and privacy issues of blockchain technology,” IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1415–1442, 2018.

[9] M. Conti, S. Kumar, C. Lal, and S. Ruj, “Blockchain security: Attack analysis and prevention strategies,” IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1352–1380, 2021.

[10] X. Fan, Y. Liu, and Z. Chen, “A blockchain-based framework for secure video forensics,” IEEE Transactions on Information Forensics and Security, vol. 17, no. 3, pp. 553–569, 2022.