

Article

BlockFECS: A Blockchain-Based Proof-of-Concept System for Metadata-Driven Evidence Correlation in Digital Forensics

Oshoke Samson Igonor ^{1,*} , Muhammad Bilal Amin ^{1,2}  and Saurabh Garg ^{1,2}¹ School of Information and Communication Technology, University of Tasmania, Hobart, TAS 7001, Australia; bilal.amin@utas.edu.au (M.B.A.); saurabh.garg@rmit.edu.au (S.G.)² School of Computing Technologies, RMIT University, Melbourne, VIC 3000, Australia

* Correspondence: oshoke.igonor@utas.edu.au

Abstract

Background/Objectives: The rapid expansion of digital evidence in modern investigations has created pressing challenges for maintaining integrity, traceability, chain of custody, and meaningful analysis across heterogeneous forensic artefacts. Conventional evidence management approaches often fall short in scalability, transparency, and the ability to correlate diverse digital evidence. This study presents BlockFECS, a blockchain-based proof-of-concept system for metadata-driven evidence correlation in digital forensics. **Methods:** BlockFECS uses Hyperledger Fabric to support auditable and tamper-resistant evidence management while capturing structured forensic metadata, including timestamps, locations, device IDs, user IDs, and file hashes. An off-chain weighted correlation algorithm assigns similarity scores between evidence pairs and classifies relationships as Related, Supplementary, Duplicate, or Unrelated. The system was evaluated using a simulated smart city accident scenario and tested for correctness, transaction latency, throughput, scalability trends, and concurrency behaviour across four computing environments. **Results:** Within the controlled proof-of-concept dataset, the correlation algorithm achieved 1.00 precision and recall for clear Related and Duplicate evidence relationships and high precision (0.90) for Supplementary relationships, although recall in this category was lower due to incomplete or noisy metadata. Performance testing showed that Create, Transfer, and Delete operations completed with sub-second latency, while correlation throughput exceeded 60 comparisons per second across all tested environments. **Conclusions:** The findings demonstrate the feasibility of combining blockchain-backed evidence integrity with lightweight metadata-driven forensic intelligence. BlockFECS contributes a proof-of-concept model for automating metadata-based evidence analysis while preserving provenance integrity and auditability, highlighting a promising direction for trustworthy and intelligent digital forensic investigation support.



Academic Editor: Ricardo Jorge Dinis-Oliveira

Received: 11 May 2026

Revised: 3 June 2026

Accepted: 12 June 2026

Published: 6 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: digital forensics; blockchain; metadata analysis; chain of custody; hyperledger fabric; evidence correlation; evidence management; smart contracts; scalability; forensic automation

1. Introduction

The proliferation of Internet of Things (IoT) devices and smart city infrastructure has vastly increased the scale and complexity of digital forensic investigations. An IoT ecosystem with billions of sensors, vehicles, and devices generates massive volumes of heterogeneous evidence data, posing serious challenges to investigators in timely and

forensically sound analysis [1,2]. Given the sensitive nature of digital forensic investigations, managing the enormous amount of data from IoT-related environments significantly complicates forensic workflows [3]. The dynamic and distributed nature of these environments presents several specific challenges, including ensuring the integrity, privacy, and provenance of digital artefacts, maintaining an auditable chain of custody (CoC), and extracting meaningful relationships from fragmented metadata. These issues are particularly acute during the Identification and Examination phases, which involve actively correlating and analyzing multiple evidence elements to reconstruct events and generate investigative leads [4–7].

To address these challenges, blockchain technology has emerged as a compelling solution for strengthening integrity, traceability, and chain-of-custody management in digital forensics, particularly where evidence must be preserved for legal or court proceedings. This effectiveness arises primarily from the inherent characteristics of the blockchain, such as decentralization, transparency, and immutability [8], enhanced security and privacy features [9], and automation capabilities through smart contracts [10]. The use of hash functions further ensures the integrity of the data [11]. These capabilities make blockchain a promising technology to ensure investigative transparency and deliver tamper-resistant CoC solutions [7,12].

However, existing blockchain-based forensic systems focus predominantly on static logging and CoC preservation, essentially using blockchain to merely “store and prove” evidence integrity. Although this approach sufficed for relatively small evidence datasets, investigators now face large and heterogeneous evidence volumes from diverse sources, including surveillance systems, embedded sensors, mobile devices, and cloud services [13]. Simply notarizing evidence on an immutable ledger is insufficient; investigators require sophisticated analytical tools without losing the integrity guarantees that blockchain provides. Current systems offer limited assistance with critical investigative tasks, such as identifying connections among artefacts or correlating events across devices, leaving these phases to manual efforts and investigator intuition. This gap indicates a critical need for blockchain-based solutions that help forensic analysts extract insights from complex, multi-source evidence sets [5].

To bridge this gap, we introduce BlockFECS (Blockchain-Enabled Forensic Evidence Correlation System), a proof-of-concept system that integrates blockchain-based evidence management with an automated metadata-driven correlation engine. BlockFECS uses a permissioned Hyperledger Fabric blockchain that securely stores critical metadata such as timestamps, geolocation, device identifiers, and file hashes. Unlike traditional systems, BlockFECS extends beyond immutable storage, utilizing a metadata-driven correlation algorithm to automatically identify semantic relationships between evidence items. The correlation logic is implemented off-chain (Node.js), while access control, evidence immutability, and audit logs are managed through smart contracts, ensuring high performance without sacrificing forensic rigor.

1.1. Key Contributions

The main contributions of this work are:

- **Blockchain-Based Forensic Framework for Evidence Correlation:** Introducing BlockFECS, which integrates metadata-weighted evidence correlation with blockchain-based evidence management. This advances beyond static evidence logging by enabling analytical support within blockchain-backed forensic infrastructures.
- **Metadata-Weighted Correlation Algorithm:** Proposing an innovative algorithm that utilizes forensic metadata (timestamps, geolocation, device identifiers, user IDs, and

file hashes) to compute weighted similarity scores, classifying evidence as Related, Supplementary, Duplicate, or Unrelated, significantly reducing manual workload.

- **Proof-of-Concept Evaluation:** Conducting a controlled evaluation using a simulated smart city vehicular accident scenario across multiple computing environments, assessing correctness, transaction latency, throughput, scalability trends, and concurrency behaviour. Results demonstrate the controlled feasibility of BlockFECS for metadata-driven evidence management and correlation, while highlighting the need for larger real-world validation before operational deployment.

The remainder of this paper is structured as follows: Section 1.2 reviews related blockchain-based forensic systems. Section 2 presents the architecture of the proposed BlockFECS system and details the metadata-based correlation approach. Section 3 describes the evaluation methodology and presents the results. Section 4 discusses qualitative benchmarking, system insights, and forensic implications, with Section 4.3 outlining the limitations of the study. Finally, Section 5 concludes the paper and discusses future research directions.

1.2. Related Work

Blockchain technology has emerged as a promising solution for improving the integrity, transparency, and traceability of digital evidence in various digital forensic domains. The literature demonstrates a dominant focus on its application to Chain-of-Custody (CoC) management, where immutability and auditability are vital requirements. This is largely attributed to the trustless and distributed characteristics of blockchain systems, which align well with legal standards for ensuring evidence authenticity and admissibility.

Several studies have emphasized the ability of the blockchain to secure the CoC process. For example, Lone and Mir [12] introduced ForensicChain, a framework using Hyperledger Composer that immutably records all evidence handling procedures, reducing the risks of human error and manipulation. Similarly, Yan et al. [14] proposed a hybrid blockchain model that combined public and permissioned layers to record forensic evidence interactions while balancing transparency and privacy.

Other notable contributions include Liu et al. [15], who implemented a digital forensics model using multidimensional hash techniques on Hyperledger Fabric. Their smart-contract-driven CoC management allowed for tamper-evident recording of evidence and its transitions. However, they noted scalability limitations due to cryptographic complexity and ledger growth over time. Li et al. [16] addressed role-based access control and transaction validation in their LEChain system by combining a consortium blockchain and CP-ABE cryptography, but reported increased latency as the number of users and access policies grew.

Despite their novelty, most of these models focus solely on record keeping, traceability, and access control, reflecting conservative use of blockchain. They treat the ledger as a passive backend and do not support real-time correlation, event reconstruction, or dynamic classification of evidence. Furthermore, most evaluations of these systems either overlook performance metrics or report latency that ranges from several seconds to more than a minute per transaction, raising questions about suitability in high-volume, time-sensitive investigations.

Although CoC has attracted substantial research interest, several studies, including the authors' previous systematic literature review (SLR) [5], highlight the underused potential of blockchain to support analytical reasoning within digital forensics. Phases such as identification and examination, which involve discovering patterns, determining relationships, and drawing conclusions, are rarely addressed by blockchain-based systems.

Akbarfam et al. [17] introduced ForensiCross, a cross-chain forensic collaboration framework that enables secure evidence exchange between institutions. However, the framework focuses primarily on interoperability and metadata protection, rather than analytical or interpretative tasks. Similarly, Nyalety et al. [18] introduced BlockIPFS, which combines IPFS for off-chain evidence storage and Hyperledger Fabric for metadata logging, but does not attempt to perform reasoning or correlation between artefacts.

Another example is the consortium blockchain presented by Chen et al. [19], where the judiciary acts as nodes on the blockchain to facilitate real-time verification. Although this begins to shift blockchain's role toward examination, it still lacks intelligent automation. Phillip and Saravanaguru [20] proposed a smart-contract-enabled traffic incident investigation framework using a hybrid blockchain-IoV. However, their work faced difficulties in scaling, especially with video-based artefacts and large volumes of vehicular sensor data, due to lack of performance optimization and reasoning logic.

Furthermore, scalability and computational overhead are recurring themes in many studies [21–24]. Cryptographic complexity, large block sizes, and consensus delays all affect performance, especially in forensic contexts where rapid ingestion and classification of multimedia or telemetry data is required. Even systems using permissioned blockchains like Hyperledger Fabric struggle with query throughput when faced with multiple evidence uploads, highlighting the need for more performance-aware designs.

Although previous attempts such as BlockSLaaS [25] have explored automated evidence recording in cloud environments, they remain focused primarily on secure logging, access control, and evidence preservation rather than context-sensitive artefact analysis. Similarly, Xiao et al. [7] demonstrated improved transaction performance in blockchain-based evidence management, but their work does not directly address metadata-driven evidence relationship classification or investigative reasoning.

More recent studies published further demonstrate that blockchain-enabled evidence management remains an active research direction. Bhattacharjee et al. [26] proposed JudicBlock, a judicial evidence preservation scheme that combines blockchain, IPFS, smart contracts, cryptographic integrity, and access control to support secure electronic law record preservation. Mbimbi et al. [27] introduced a private blockchain-driven digital evidence management framework using collaborative mining and NFT-based representation to improve evidence authenticity, integrity verification, and scalability in law-enforcement evidence management systems. Mahajan et al. [28] presented ChainSEAL, a hybrid blockchain-IPFS evidence ledger that stores encrypted evidence files off-chain while anchoring hashes, metadata, and custody events on-chain for confidentiality, traceability, and legal admissibility. From a broader digital-forensics automation perspective, García Fernández et al. [29] proposed an agentless and automated framework for acquiring digital evidence in heterogeneous corporate, cloud, and industrial networks, reinforcing the growing need for scalable, traceable, and minimally intrusive forensic workflows. In addition, Haji et al. [30] examined blockchain-based chain of custody and digital evidence legality in post-conflict prosecutions, emphasizing that blockchain can support provenance, continuity, and auditability but should not replace judicial assessment or the right to challenge evidence. Collectively, these recent works strengthen the case for blockchain-supported evidence integrity and digital forensic automation, but they remain primarily focused on evidence preservation, acquisition, access control, storage efficiency, legal admissibility, or custody management rather than metadata-driven classification of relationships between heterogeneous evidence artefacts.

Overall, the reviewed studies show that blockchain has been applied mainly as a mechanism for preserving evidence integrity, recording chain-of-custody events, supporting access control, and improving traceability. These contributions are important because

they address core requirements for the admissibility and reliability of digital evidence. However, most existing systems remain centred on evidence storage, logging, verification, and custody management. Comparatively fewer studies examine how blockchain-supported infrastructures can assist the Identification and Examination phases of digital forensics, where investigators need to identify relationships between heterogeneous artefacts, reduce redundancy, and prioritise evidence for further analysis. In addition, many existing frameworks provide limited empirical evaluation, are tested on narrow scenarios, or do not examine scalability under different deployment conditions. These shortcomings indicate that there remains a need for blockchain-supported forensic systems that combine trustworthy evidence management with transparent and reproducible metadata-based analytical support.

2. Materials and Methods

2.1. Overview

The primary scope of this study is to extend the existing use of blockchain in digital forensics beyond secure, privacy-aligned, and transparent Chain-of-Custody (CoC) management. While CoC remains essential for preserving evidence integrity and admissibility, digital investigations also require mechanisms that can support evidence interpretation, relationship discovery, redundancy reduction, and investigative prioritisation. This study therefore focuses on metadata-based evidence correlation as a way to support the Identification and Examination phases of digital forensics.

To address this objective, this study proposes the Blockchain Forensics Evidence Correlation System (BlockFECS), a Hyperledger Fabric-based proof-of-concept framework that combines blockchain-backed evidence management with an off-chain metadata correlation module. The blockchain layer supports evidence integrity, access control, provenance, and auditability, while the off-chain correlation module assesses relationships between new and existing evidence using forensic metadata such as timestamps, geolocation, device identifiers, user identifiers, and file hashes.

By separating blockchain-backed custody management from off-chain analytical processing, BlockFECS aims to preserve the forensic strengths of blockchain while avoiding the performance burden of executing computational correlation directly on-chain. The following subsections describe the system architecture, evidence data model, smart-contract functions, off-chain correlation workflow, and evidence storage design.

A conventional permission-controlled database with audit logs could support several evidence management functions, including access control, record storage, and event logging. However, such systems typically depend on administrative trust in the database owner and privileged users, which can create concerns about unilateral record modification, audit-log control, and cross-organisational accountability in forensic settings. Prior studies have noted that centralised digital evidence or judicial record management systems may be vulnerable to tampering, unauthorised access, single points of failure, weak auditability, lack of transparency, and poor traceability across the chain of custody [26–28]. Blockchain-based forensic evidence systems have therefore been proposed to strengthen integrity, transparency, access control, custody tracking, and tamper-evident verification by combining cryptographic hashing, smart contracts, distributed ledgers, and off-chain storage for larger evidence files [26–28]. BlockFECS therefore uses a permissioned blockchain not because blockchain is universally superior to databases, but because Hyperledger Fabric supports a tamper-evident shared ledger, distributed validation, role-aware access control, and a common verifiable audit trail across participating organisations. This makes blockchain particularly suitable where provenance, cross-organisational accountability, and resistance to unilateral record modification are important forensic requirements.

2.2. The BlockFECS Framework

2.2.1. Research Tools

This study employed several tools to develop, test, and evaluate the BlockFECS prototype:

- Hyperledger Fabric: Selected based on its widespread use in the reviewed blockchain-forensics literature and its features such as privacy controls, role-based access, permissioned participation, and smart contract support [5]. It is a permissioned blockchain framework that supports secure and traceable handling of digital evidence without requiring a native cryptocurrency [31].
- Smart Contracts in Go: Used for implementing the on-chain evidence management logic. Go was chosen because of its performance, strong concurrency support, and native compatibility with Hyperledger Fabric chaincode development [32].
- Node.js and React.js: Node.js was used to implement the backend API layer and to interface with the Hyperledger Fabric network through SDK-based communication. React.js was used to implement the frontend dashboard, enabling investigators to upload evidence, view evidence records, and interact with the correlation results.
- Google Cloud API: Used to support off-chain evidence file storage during the proof-of-concept implementation. Evidence files were stored off-chain while their SHA-256 hashes and associated metadata were recorded on the blockchain. This design reduced on-chain storage overhead while preserving integrity verification through hash comparison.
- Python: Python was used in this study for post-processing, statistical analysis, regression analysis, and visualisation of the experimental results. Specifically, the pandas and NumPy libraries were used for data cleaning and numerical analysis, while matplotlib and seaborn were used to generate the evaluation figures.

2.2.2. System Design and Implementation

The BlockFECS prototype is structured as a layered platform. It anchors the proof of integrity on a Hyperledger Fabric blockchain while offloading large media files to cloud storage and conducting metadata correlation in a separate analytical layer.

System Architecture

The proposed architecture combines permissioned blockchain infrastructure and a metadata-driven automation layer to support secure evidence lifecycle management and automatic correlation. As shown in Figure 1, the BlockFECS architecture includes forensic stakeholders such as First Responders, Second Investigators, and authorised external actors, including law enforcement and legal professionals. Evidence items are treated as blockchain assets with key metadata: timestamp, geolocation, device ID, user ID, and file hash. Role-based access control (RBAC) ensures actions (e.g., create, transfer, delete) are permitted only based on user roles. These actions are executed as smart contracts and recorded immutably via blockchain consensus, promoting transparency and traceability [12,33].

Smart Contract Model

The chaincode extends the simple asset-transfer template by embedding correlation metadata and expanding data fields relevant to forensic correlation. Note that the `CorrelateEvidence` functionality is implemented off-chain to reduce the blockchain load and improve the responsiveness of the system. The smart contract focuses on the core evidence lifecycle actions, while the correlation logic is located in a dedicated back-end service. Access control is enforced in the smart contract by checking the client Membership Service Provider Identifier (MSP ID) in Hyperledger Fabric before state updates. In an operational deployment, compromised investigator credentials would need to be handled through

Hyperledger Fabric's identity management and revocation mechanisms. Investigator identities are issued through a Fabric Certificate Authority and validated by the Membership Service Provider (MSP). If an investigator's credentials are compromised, the corresponding certificate can be revoked by the issuing Certificate Authority and distributed through a certificate revocation list. Once the network configuration and revocation information are updated, peers and ordering services can reject transactions signed using revoked credentials. In a forensic setting, this technical revocation process should be combined with operational controls such as immediate account suspension, key rotation, audit review of recent transactions, and re-issuance of credentials only after identity re-verification. Transaction types and role-based access control policies are detailed in Table 1.

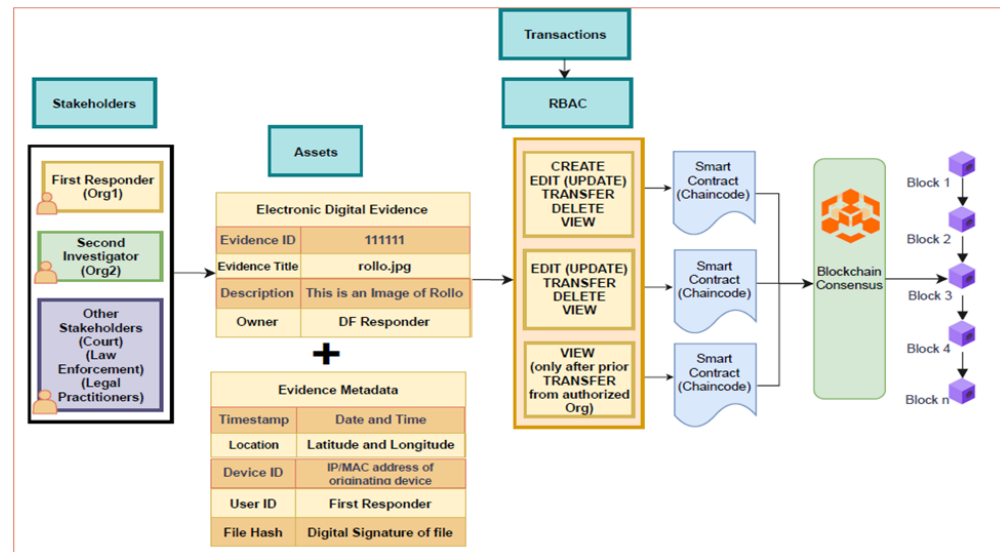


Figure 1. Simplified Architecture of the Proposed BlockFECS Framework.

Table 1. Smart Contract Transactions and Access Control Policies.

Transaction Name	Purpose	User Role
CreateEvidence	Hashes file and stores metadata	Org1 Only
TransferEvidence	Updates ownership and geolocation logs	Org1, Org2
EditEvidence	Allows title/description edits	Org1, Org2
DeleteEvidence	Marks status as deleted	Current Owner
QueryEvidenceByCaseID	Fetches all evidence items associated with a specified Case ID	All Investigators
GetEvidenceDetails	Fetches individual evidence items	All Investigators
GetAllEvidence	Fetches all evidence items	All Investigators
GetTransactionLogs	Returns ordered transaction logs	All Investigators

Appendix A provides pseudocode for the smart contract transactions and query functions listed in Table 1.

Backend Service Design and Evidence Storage

To streamline back-end interactions with the blockchain, the system maintains a long-lived Fabric Gateway object that is refreshed periodically to avoid repeated TLS handshakes. Back-end REST controllers handle validation, orchestrate evidence uploads and blockchain transactions, and return appropriate JSON responses to the front-end.

When an investigator uploads a file via the React UI, a multipart request is sent to the `/api/evidence/upload` endpoint. The backend then streams the file to an off-chain cloud storage location, computes its SHA-256 hash in real time, and records the metadata on the blockchain by invoking the `CreateEvidence` smart contract. Since only the cryptographic

digest and metadata are stored on-chain, the system preserves storage efficiency while maintaining forensic guarantees.

The correlation endpoint `/api/evidence/correlation/:id` triggers a search for all stored evidence, evaluates pairwise similarity scores based on forensic metadata (timestamp = 0.4, location = 0.3, device = 0.15, user = 0.1, file hash = 1.0 for exact match) and identifies related artefacts using a defined threshold. This modular architecture ensures performance scalability without compromising data integrity or auditability.

Rather than embedding computationally intensive logic directly into smart contracts, the correlation functionality is implemented off-chain in a dedicated correlation layer. The blockchain layer therefore provides integrity, provenance, access control, and auditability, while the off-chain layer performs the computational comparison required for metadata-based evidence correlation. This decision offloads analytical processing from the blockchain, maintaining performance while preserving forensic properties such as transparency and resistance to tampering [34–36].

As shown in Figure 2, the correlation module is activated after new evidence is submitted. The backend retrieves relevant evidence metadata from the blockchain and passes it to the off-chain correlation engine, where the incoming evidence is compared with existing evidence records using the weighted similarity algorithm described in Section 2.3. The resulting relationship label, similarity score, and matched-metadata reasons are returned to the backend and frontend for audit-aware display to the investigator.

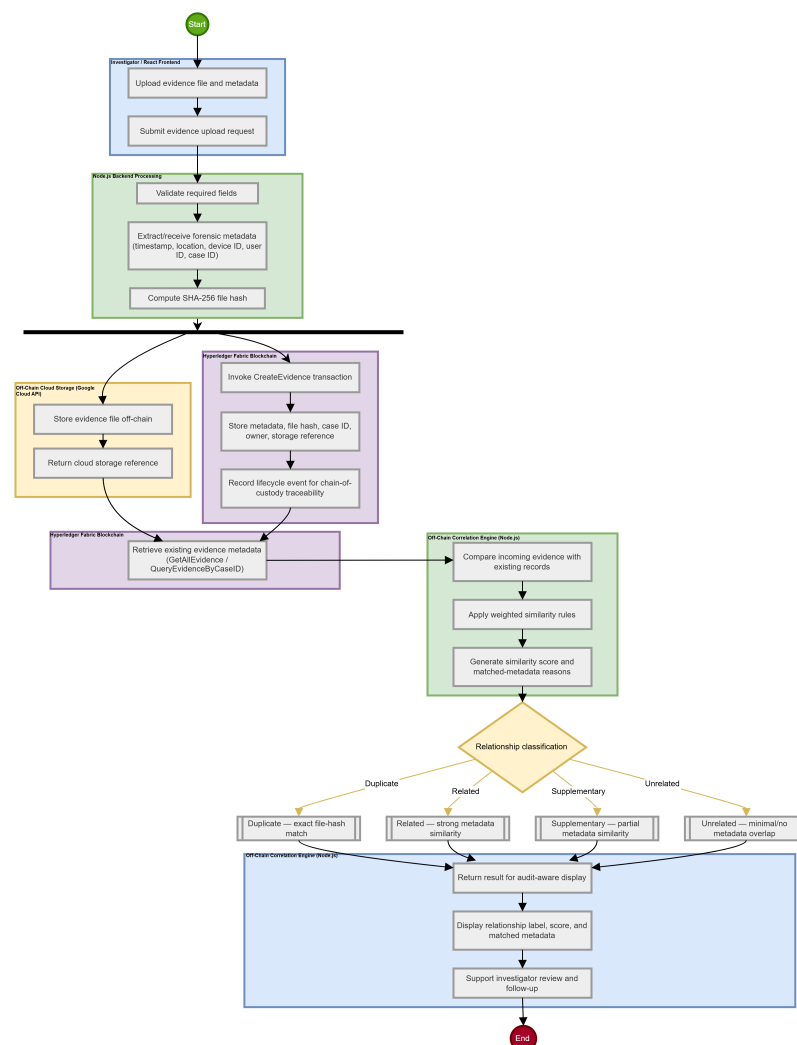


Figure 2. Metadata-driven evidence correlation workflow in BlockFECS.

2.3. The Evidence Correlation Algorithm

Digital evidence originating from heterogeneous IoT sensors is often noisy, partially missing, or timestamp skewed. To remain court-admissible, we propose a transparent rule-based correlation algorithm in which (i) each metadata field is compared deterministically, (ii) a weighted similarity score is produced, and (iii) the final categorical label and correlation summary are generated off-chain and can be persisted back to the blockchain for auditability. The logic purposely echoes the way human investigators triage evidence while remaining machine-executable and reproducible, satisfying the “transparent, explainable, chain of evidence” criteria highlighted by Lone and Mir [12]. The algorithm also applies configurable tolerances for both time ($\pm\Delta t$) and location (Δm , via the haversine distance) to acknowledge real-world uncertainties such as clock skew, drift, time zone offsets, environmental factors, and even deliberate tampering [37]. The pseudo-code for the correlation metadata-driven algorithm is outlined in Algorithm 1.

Algorithm 1 Metadata-Weighted Cross-Case Correlation

Require: E_target (incoming evidence), $allEvidence[]$ (ledger entries)
Ensure: $matchList[]$ {id, score, label, reasons}

- 1: Define weights: $w_t = 0.40, w_l = 0.30, w_d = 0.15, w_u = 0.10, w_h = 1.00$
- 2: Define tolerances: $\Delta t = 3 \text{ min}, \Delta m = 30 \text{ m}$
- 3: Initialize $matchList = []$
- 4: **for all** $E \in allEvidence$ **do**
- 5: **if** $E.evidenceID == E_target.evidenceID$ **then**
- 6: continue
- 7: **end if**
- 8: Compute $simTime \leftarrow 1$ if $|E.timestamp - E_target.timestamp| \leq \Delta t$ else 0
- 9: $dist \leftarrow \text{haversine}(E.lat, E.lon, E_target.lat, E_target.lon)$
- 10: $simLoc \leftarrow \max(0, 1 - dist/\Delta m)$
- 11: $simDev \leftarrow 1$ if $E.deviceId == E_target.deviceId$ else 0
- 12: $simUser \leftarrow 1$ if $E.userId == E_target.userId$ else 0
- 13: $simHash \leftarrow 1$ if $E.fileHash == E_target.fileHash \wedge E.fileHash \neq ""$ else 0
- 14: $score \leftarrow w_t \cdot simTime + w_l \cdot simLoc + w_d \cdot simDev + w_u \cdot simUser + w_h \cdot simHash$
- 15: **if** $simHash == 1$ **then**
- 16: label $\leftarrow$ “Duplicate”
- 17: **else if** $score \geq 0.70$ **then**
- 18: label $\leftarrow$ “Related”
- 19: **else if** $score \geq 0.40$ **then**
- 20: label $\leftarrow$ “Supplementary”
- 21: **else**
- 22: continue
- 23: **end if**
- 24: Initialize $reasons = []$
- 25: **if** $simHash$ **then**
- 26: reasons.append(“exact file hash match”)
- 27: **end if**
- 28: **if** $simTime$ **then**
- 29: reasons.append(“timestamps within 3 min”)
- 30: **end if**
- 31: **if** $simLoc > 0.8$ **then**
- 32: reasons.append(“locations within 30 m”)
- 33: **end if**
- 34: **if** $simDev$ **then**
- 35: reasons.append(“same device ID”)
- 36: **end if**
- 37: **if** $simUser$ **then**
- 38: reasons.append(“same user ID”)
- 39: **end if**
- 40: Append {id: $E.evidenceID$, score, label, reasons} to $matchList$
- 41: **end for**
- 42: **return** $matchList$

2.3.1. Weighted Similarity Scheme

To effectively translate heterogeneous IoT metadata into a reliable correlation score, the system uses a fixed weighted scheme applied to each metadata attribute. This approach balances the forensic logic with the practical implementation requirements by reflecting the relative forensic value of each metadata type. Rather than privileging a single attribute (except file hash), the scheme ensures that evidence is judged holistically across multiple cues.

Table 2 outlines the five key metadata attributes and their corresponding weights used in the similarity algorithm. The weight assignment is based on their forensic relevance, robustness to manipulation, and likelihood of presence between devices. Temporal and spatial cues receive the highest weight because of their reliability and frequent availability across digital artefacts. The file hash receives full weight, as it confirms byte-for-byte duplication. Device ID and User ID, though informative, are weighted lower due to their variability, spoofing risks, or possible absence in many IoT devices. The weights used in this study should be interpreted as literature-informed, scenario-specific values for the selected smart city vehicular accident context rather than universal forensic weights. In this scenario, timestamp and geolocation receive higher weights because temporal proximity and spatial proximity are central to reconstructing IoT-based traffic incidents. However, other investigative contexts may require different weight distributions. For example, user identity, account activity, device provenance, network logs, or cloud service metadata may carry greater importance in insider-threat, cyber intrusion, mobile, or cloud forensic investigations. Therefore, the current weighting scheme is intended to demonstrate the feasibility of transparent metadata-driven correlation, while future versions should support configurable, context-aware, or data-driven weighting.

Table 2. Metadata Attributes and Assigned Weights for Evidence Correlation in IoT Forensics.

Metadata Attribute	Weight	Rationale
Timestamp (± 3 min)	0.40	Reliable clue for sequence reconstruction; accounts for delays, network jitter, or unsynchronized clocks [37–39].
Geolocation (≤ 30 m)	0.30	Crucial for vehicular contexts; accounts for GPS drift [40].
Device ID	0.15	Identifies source hardware; moderate role in provenance.
User ID	0.10	Operator context; weighted low due to spoofing risks [41].
File Hash	1.00	Definitive byte-for-byte match; highest possible weight.

These weight values reflect best practices in the digital forensics literature, particularly those focusing on event reconstruction, timeline analysis, and device-based evidence sourcing, especially in the case context of IoT vehicular forensics. For example, timestamps receive the largest share (0.40) because timeline reconstruction is routinely the first analytic step and remains fragile in the face of clock skew, drift, and tampering; investigators therefore rely on a short tolerance window rather than exact equality [37]. Spatial proximity is next (0.30) because location traces from mobile and IoT sensors are highly central to linking people and devices to crime scenes [38,40]. However, device IDs carry less influence (0.15) because repeated digital IDs such as VINs and ECU identifiers or camera serial numbers have proved valuable for establishing common provenance in incidents in the past [42]. Finally, User IDs have the lowest weight because, although they may help reconnect devices associated with a user or network identity, such identifiers can be spoofed [41] and may be absent in many IoT devices such as CCTV cameras and vehicle sensors.

2.3.2. Thresholds

To classify evidence relationships, BlockFECS uses threshold values that map similarity scores to categorical labels. These thresholds guide how automated correlation outcomes

are interpreted and help investigators quickly distinguish strong, weak, or irrelevant matches. BlockFECS classifies evidence relationships based on the computed similarity score as follows:

- Duplicate (score ≥ 1.00): Indicates an exact file hash match, confirming that the same artefact has been uploaded more than once.
- Related (score between 0.70 and 0.99): Suggests a strong contextual match across multiple metadata fields such as timestamp, location, and device ID.
- Supplementary (score between 0.40 and 0.69): Represents partial similarity that may be useful for further investigation; merits manual review.
- Unrelated (score < 0.40): Indicates minimal or no overlap; the artefact is likely not relevant to the current case context.

2.3.3. Correlation Algorithm Logic

To support intelligent evidence analysis within digital forensics, BlockFECS introduces a metadata-weighted correlation algorithm that classifies relationships between an incoming evidence item and all existing records in the ledger. The system evaluates similarity across five metadata attributes: timestamp, location, device ID, user ID, and file hash. Scores are computed using field-specific comparisons and then aggregated using the weighted scheme in Table 2. If the similarity score meets or exceeds a predefined threshold, a classification label and explanations are generated.

The current implementation performs pairwise metadata comparison between a target evidence item and existing evidence records. For a single incoming evidence item, this requires comparison with N existing records, resulting in $O(N)$ complexity per submission. However, when all pairwise relationships in a dataset are evaluated, the total comparison cost becomes $O(N^2)$. While this is acceptable for the controlled proof-of-concept dataset used in this study, it may become inefficient as evidence repositories grow. Therefore, large-scale deployments would require candidate filtering, indexing by case ID, time window, geolocation, or device identifier, batching strategies, or graph-based correlation to reduce the comparison space before applying the weighted similarity algorithm.

The pseudo-code for the correlation function (`CorrelateEvidence`) is outlined in Algorithm 1. It iteratively compares the incoming evidence against existing items, computes a cumulative score, and stores human-readable reasons for any matched attributes. This correlation algorithm logic is encoded to run off-chain, while the resulting match list can be persisted through the blockchain-supported `CorrelateEvidence` workflow for auditability.

The proposed correlation method should be distinguished from advanced IoT anomaly detection and machine learning-based forensic analysis approaches. IoT anomaly detection methods typically aim to identify abnormal device behaviour, attacks, or deviations from expected operational patterns in IoT data streams, such as anomalous communication patterns or botnet-related traffic from compromised devices [43,44]. In contrast, BlockFECS focuses on post-collection forensic evidence correlation. Its purpose is to classify relationships between evidence items using transparent metadata cues after evidence has been registered in the system. This rule-based design supports interpretability and auditability, which are important for forensic reasoning, but it does not provide adaptive learning, behavioural modelling, or automatic discovery of unknown attack patterns. Therefore, the proposed method should be viewed as a transparent baseline for metadata-driven evidence relationship classification rather than a replacement for advanced anomaly detection. Future versions could integrate machine learning or anomaly detection methods to support dynamic weighting, candidate filtering, and detection of more complex evidence relationships.

2.4. Chain of Custody

To establish a legally defensible and verifiable Chain of Custody (CoC), BlockFECS records all evidence lifecycle actions in a tamper-evident manner and logs actions executed through smart contracts into a structured `TransactionLog` asset. These logs are recorded on-chain, making them tamper-evident and resistant to unauthorised modification. Each log entry includes the action type, timestamp, associated evidence ID, role of the user performing the action, and (if applicable) both the old and new values for edited fields. This ensures complete traceability of all actions performed on evidence.

Logging occurs across three architectural layers:

- Chaincode (`simple-contract.go`): Functions like `logSimpleTransaction` and `logEditTransaction` generate human-readable log entries written into the Fabric world state.
- Backend (`evidenceController.js`): A REST API endpoint `/api/evidence/logs` queries the chain and structures entries for UI rendering.
- Frontend (React Dashboard): Users can access and export logs via a formatted table for court submission.

2.5. Front-End Interaction and Evidence Upload

The front-end application, developed using React.js, is designed to facilitate intuitive interaction between investigators and the blockchain-based forensic system. The core application state is managed through React Context, ensuring consistent access across interface components. Evidence entries are rendered within a paginated table supporting full-text search, column-level filtering, and dynamic sorting.

2.5.1. Evidence Upload Process

When uploading evidence, investigators interact with a modal form capturing metadata and file attachments. The backend streams the file to off-chain cloud storage via the Google Cloud API and simultaneously computes its SHA-256 cryptographic hash. This 64-byte digest, along with the associated metadata, is passed to the Hyperledger Fabric chaincode via the `CreateEvidence` transaction, ensuring forensic integrity without bloating the on-chain ledger.

Although IPFS is a popular off-chain storage option in blockchain-based forensic systems because it supports content-addressed and decentralised file referencing, this proof-of-concept used cloud-based storage through the Google Cloud API for pragmatic implementation reasons. First, the primary focus of this study was to evaluate blockchain-backed metadata integrity, chain-of-custody logging, and metadata-driven evidence correlation rather than decentralised file-storage performance. Second, cloud storage provided stable APIs, mature access-control mechanisms, predictable upload and retrieval behaviour, and straightforward integration with the Node.js and React prototype. Third, this choice reduced implementation complexity and helped isolate blockchain transaction and correlation performance from IPFS-specific factors such as pinning, node availability, replication policy, and distributed retrieval latency.

However, IPFS remains a suitable alternative for decentralised forensic deployments, particularly where content-addressed storage, decentralised availability, and reduced reliance on a single cloud provider are required. Recent systems such as JudicBlock [26] and ChainSEAL [28] demonstrate the continuing relevance of blockchain-IPFS architectures for evidence preservation and secure evidence ledgers. Therefore, future versions of BlockFECS could integrate IPFS or a hybrid cloud-IPFS storage model and compare storage availability, retrieval latency, confidentiality controls, and forensic auditability across different off-chain storage designs.

2.5.2. Correlation Summary and Insights

The interface provides a correlation summary table allowing investigators to view computed relationships, explore the rationale behind classifications (e.g., “Related” or “Duplicate”), and initiate further investigative actions. This balances usability with forensic rigor for nontechnical users while maintaining evidentiary integrity.

3. Results

This section rigorously evaluates the BlockFECS prototype from two complementary angles: the correctness of its metadata-driven correlation algorithm and the operational performance of the underlying blockchain evidence platform.

3.1. Correctness Evaluation of the Correlation Algorithm

Detecting the accurate relationship between digital evidence is critical in forensic investigations. The correctness of the correlation algorithm was evaluated using a simulated smart city accident scenario with heterogeneous sources of IoT evidence, including CCTV, dashcams, vehicle sensors and smartphones. Each evidence item was tagged with metadata: timestamp, location, device ID, user ID, and file hash. The goal was to validate whether the algorithm could infer relationships using only these metadata.

3.1.1. Dataset Preparation

The dataset was adapted from the publicly available TON_IoT dataset [45]. Since this dataset does not provide structured forensic metadata required for evidence correlation, synthetic forensic fields were introduced to simulate realistic digital investigation conditions. This approach was adopted due to the limited availability of open, standardised IoV forensic datasets containing evidence-level timestamps, geolocation, device identifiers, user identifiers, hashes, and case identifiers. Therefore, the adapted dataset was used as a controlled proof-of-concept testbed for evaluating the proposed correlation logic rather than as a definitive real-world forensic benchmark. The synthetic metadata values were assigned according to the smart city accident scenario, with related records designed to share plausible temporal, spatial, device, user, or hash-based relationships, while unrelated and noisy records were introduced to test the algorithm’s ability to distinguish weak or misleading similarities. These additions included:

- `evidenceID`—Unique identifier per artefact
- `userId`—Logical user/witness identifier
- `deviceId`—Capturing device ID
- `fileHash`—SHA-256 cryptographic hash
- `googleDriveId`—Cloud file storage ID placeholder
- `caseId`—Investigation group tag

These metadata additions aligned the dataset with the schema of the system. To support correlation testing, fuzzy thresholds were implemented: timestamps within ± 3 min and locations within 30 m were considered to match [37].

Stress Testing

To simulate non-ideal forensic conditions, a custom script injected controlled noise:

- **Missing Fields:** Randomly removed 20% of metadata fields.
- **Near Misses:** Slightly shifted timestamp (4–6 min) and location (31–50 m).
- **Deceptive Similarity:** Inserted unrelated records with overlapping metadata.
- **Shuffled Case IDs:** Broke implicit grouping.
- **Adversarial Noise:** Injected spoofed or tampered metadata.

3.1.2. Experimental Setup

The final dataset contained 89 evidence items, generating 4186 unique pairwise comparisons. The analysis was performed using Python 3.11 in Google Colab. Four system environments were used for testing. Table 3 summarises the hardware configurations used to evaluate the prototype across low-resource, moderate, and higher-capacity deployment settings.

Table 3. System Specifications for Evaluation.

System	RAM	vCPUs	Disk
System 1	4 GB	2	30 GB
System 2	8 GB	4	30 GB
System 3	16 GB	8	30 GB
System 4	32 GB	16	30 GB

The four system configurations were selected to evaluate how BlockFECS behaves across different resource levels rather than to claim production-scale deployment readiness. System 1 represents a low-resource environment that may resemble a lightweight field, edge, or constrained laboratory setup. Systems 2 and 3 represent progressively stronger workstation-style environments, while System 4 represents a higher-capacity server-style configuration. This staged configuration allows the evaluation to examine whether latency, throughput, scalability, and concurrency behaviour improve as CPU and memory resources increase. The purpose of using four systems was therefore to assess resource sensitivity, performance trends, and controlled deployment feasibility across practical hardware tiers.

Evaluation Metrics

The following metrics were used to assess the algorithm's accuracy:

$$\text{Precision} = \frac{\text{Correctly Identified}}{\text{Correctly Identified} + \text{Incorrectly Identified}} \quad (1)$$

$$\text{Recall} = \frac{\text{Correctly Identified}}{\text{Correctly Identified} + \text{Missed Detections}} \quad (2)$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Ground Truth Methodology

Ground truth was established using a hybrid strategy to reduce reliance on a single source of judgement. The purpose of this process was to compare the algorithm's output against independently reviewed relationship labels derived from the scenario logic and metadata patterns.

- **Manual Labeling:** A subset of evidence pairs was manually analyzed by the author using the scenario context and metadata fields to establish definitive reference labels.
- **Test Script:** A separate test script was used to generate rule-based labels for internal consistency checking. These labels were not treated as the sole ground truth because they were derived from logic similar to the algorithm under evaluation.
- **Advanced LLM-Driven Analysis:** To support scalable review of evidence-pair relationships, OpenAI's ChatGPT-4.1 was used to assist in labelling relationships based solely on metadata. The LLM-assisted labelling was used as a supplementary consistency-checking mechanism rather than as a substitute for forensic validation. The LLM

was provided only with the relevant metadata fields and scenario rules, not with the algorithm's predicted output, to reduce confirmation bias during the labelling process.

The final labels were reviewed against the scenario design and manually inspected examples to reduce self-validation bias. Importantly, the LLM-assisted review was not treated as an authoritative forensic ground truth, and it was not provided with the algorithm's predicted outputs during labelling. This reduced the risk of confirmation bias, although it does not remove the need for independent expert validation. Therefore, the reported correctness results should be interpreted as proof-of-concept evidence, and future work should validate the algorithm using independently labelled real-world forensic datasets reviewed by qualified forensic practitioners. Table 4 presents representative evidence-pair examples used to illustrate how different metadata overlaps produced Related, Supplementary, Duplicate, and Unrelated classifications.

Table 4. Sample Evidence Pairs and Correlation Scores.

Evid. A	Evid. B	Score	Category	Matched on
EVID006	EVID027	0.95	Related	Timestamp, Location, Device, User
EVID009	EVID013	0.95	Related	Timestamp, Location, Device, User
EVID031	EVID032	0.40	Supplementary	Timestamp only
EVID015	EVID030	1.95	Duplicate	File Hash
EVID001	EVID004	0.00	Unrelated	nil

These examples show how the correlation score changes according to the number and strength of matched metadata attributes. Evidence pairs sharing timestamp, location, device, and user metadata achieved high Related scores, while pairs with only partial overlap were classified as Supplementary. Exact file-hash matches were classified as Duplicate, and pairs with no meaningful metadata overlap were treated as Unrelated.

3.1.3. Correlation Classification Results

Table 5 summarises the precision, recall, and F1-score results for each evidence relationship category across the 4186 evaluated evidence pairs.

Table 5. Performance Metrics of the Correlation Algorithm by Category (4186 Pairs Evaluated).

Category	Correct	Incorrect	Missed	Precision	Recall	F1
Related	3	0	0	1.00	1.00	1.00
Supplementary	196	21	952	0.90	0.17	0.29
Duplicate	55	0	0	1.00	1.00	1.00
Unrelated	2959	952	21	0.76	0.99	0.86

As depicted in Figure 3, the algorithm output 1.00 precision and recall in the *Related* and *Duplicate* categories, indicating strong performance where metadata signals were complete and highly consistent. In the *Supplementary* category, it achieved high precision (0.90) but substantially lower recall (0.17), indicating a conservative approach that avoids false positives at the cost of missing some subtle or incomplete relationships. This result suggests that the current rule-based weighting scheme is reliable for clear relationships but less effective when evidence contains partial, noisy, or missing metadata. The low recall for the Supplementary category is an important limitation because Supplementary relationships often represent weak but potentially useful investigative leads. In the current implementation, the threshold range for Supplementary evidence favours precision over recall, meaning that the system avoids excessive false positives but may miss weaker relationships when metadata is incomplete or noisy. This design choice was appropriate for a

conservative proof-of-concept forensic setting, where false positives can increase investigator workload and risk misleading interpretation. However, operational deployments would require threshold tuning, sensitivity analysis, and context-aware weighting to improve the detection of Supplementary relationships without substantially increasing false positives.

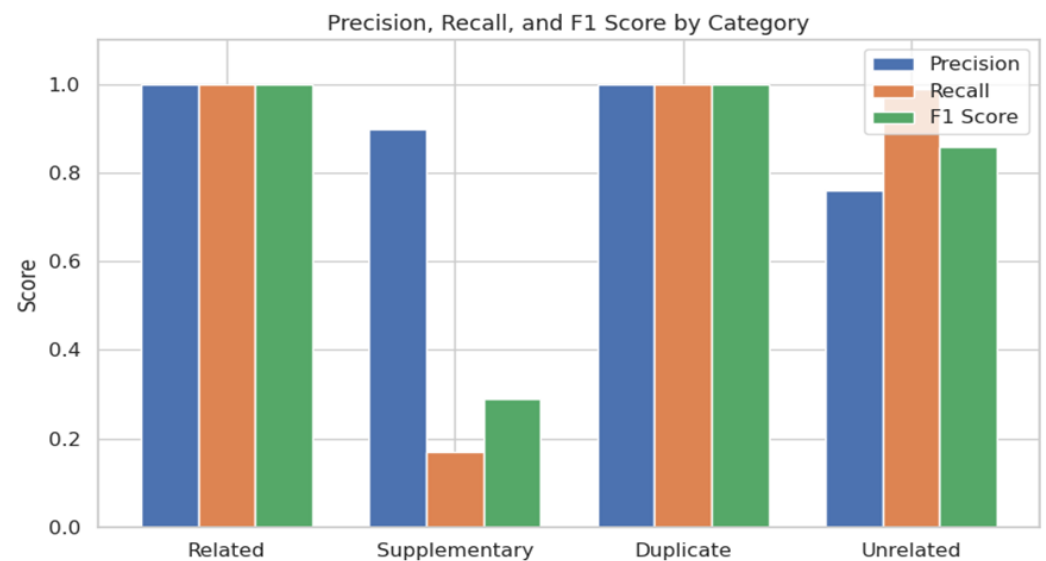


Figure 3. Precision, Recall, and F1 Score by Evidence Correlation Category.

Because the evaluation dataset was constructed using controlled scenario-based metadata, the precision and recall values should be interpreted as proof-of-concept correctness results rather than as estimates of real-world forensic accuracy. The synthetic relationships make it possible to test whether the correlation logic behaves as expected under known conditions, but they may also introduce bias because the relationship structure is partly determined by the same metadata attributes used by the algorithm. This limitation reinforces the need for future validation using independently labelled, real-world forensic datasets with naturally occurring noise, missingness, and ambiguous evidence relationships.

3.2. Performance Evaluation of the Blockchain System

The blockchain platform was evaluated to determine how core on-chain operations, including creating, transferring, deleting, and correlating evidence records, perform across the four hardware configurations. Controlled scalability and concurrency tests were run on the system profiles described in Table 3. Table 6 presents the average latency and throughput results for the core blockchain operations across 30 trials.

Table 6. Average Latency (s) and Throughput (TPS) for Core Transactions Across 30 Trials.

System	Create Time (s)	Create TPS	Transfer Time (s)	Transfer TPS	Delete Time (s)	Delete TPS	Correlation Time (s)	Correlation TPS
System 1	0.451	2.270	0.443	2.325	0.460	2.243	0.016	63.025
System 2	0.433	2.364	0.442	2.287	0.427	2.359	0.015	68.002
System 3	0.416	2.452	0.412	2.460	0.386	2.642	0.015	70.257
System 4	0.417	2.467	0.404	2.512	0.407	2.498	0.014	72.578

3.2.1. Baseline Performance Results

The 30-trial baseline results show that all core blockchain operations completed with sub-second average latency across the four tested environments. This indicates that the prototype can support basic evidence lifecycle operations without introducing substantial

transaction delay under controlled sequential workloads. The throughput results also show that performance generally improved as system resources increased, with System 4 recording the highest throughput across the evaluated operations.

The relatively similar results observed for Systems 1, 2, and 3 suggest that, under small sequential workloads, the measured performance was influenced more by Hyperledger Fabric transaction-submission, endorsement, ordering, and commit overhead than by local CPU and memory capacity. Since the baseline experiment did not impose heavy concurrent demand, additional RAM and vCPU resources did not produce proportional improvements at this stage. This finding implies that lightweight deployments of BlockFECS may be feasible for routine evidence registration and custody operations, provided that workloads remain moderate and mostly sequential.

However, the clearer performance differences observed in the scalability and concurrency experiments indicate that hardware resources become more important as evidence volume, batch size, and simultaneous request rates increase. From a research perspective, this suggests that BlockFECS, as a proof-of-concept system, demonstrates controlled operational feasibility rather than full validation as a large-scale forensic platform. Future deployments would require workload-aware optimisation, candidate filtering for correlation queries, batching strategies, and distributed infrastructure to support high-volume digital forensic environments.

3.2.2. Discussion

The performance evaluation results reinforce the viability of the blockchain-based forensic system in diverse hardware configurations. All four systems exhibited consistent and predictable latency patterns for each core operation, demonstrating deterministic smart contract execution under typical load conditions. The `correlateEvidence` function stood out, achieving sub-20 ms latencies and throughput exceeding 60 TPS in all systems.

3.3. Scalability Evaluation of the Blockchain-Based Evidence System

Scalability testing is vital for assessing how well a system performs as demand increases. This section presents the results of a detailed scalability analysis of the developed forensic evidence system in four different computational environments.

3.3.1. Test Environment and Setup

The hardware configurations remain the same as in Table 3. The operations were triggered sequentially in batch sizes of 10, 50, 100, 500, 1000, and 2000 records. Linear regression models were programmatically fitted to the per-record times to evaluate performance scaling using the format $y = mx + c$.

3.3.2. Scalability Results

Aggregate Time Metrics Across Systems

Table 7 summarizes the minimum, maximum, and mean operation times recorded across all tested batch sizes.

Table 7. Min, Max, and Mean Operation Times (in Seconds) Across All Systems.

System	Create Time (s)			Transfer Time (s)			Avg Correlation Time (s)		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
System 1	4.497	871.43	264.70	4.431	856.41	260.86	0.015	0.272	0.088
System 2	4.308	850.14	259.05	4.165	835.01	254.39	0.015	0.248	0.081
System 3	4.772	828.51	256.49	4.286	830.41	255.91	0.015	0.221	0.081
System 4	4.889	820.31	252.90	4.236	833.11	254.39	0.012	0.231	0.074

Figures 4–6 visualise how total operation time changes as batch size increases across the four system configurations. These figures complement Table 7 by showing the growth pattern of Create, Transfer, and Correlation operations across increasing workload sizes.

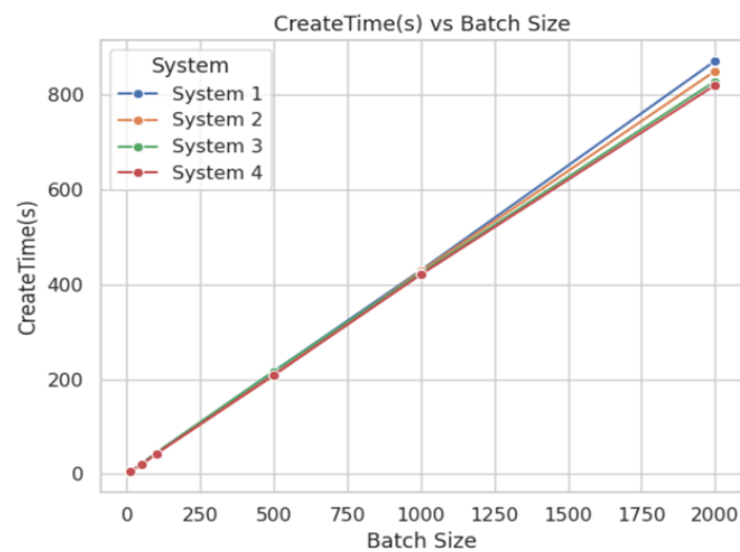


Figure 4. CreateTime (s) vs. Batch Size Across System Configurations.

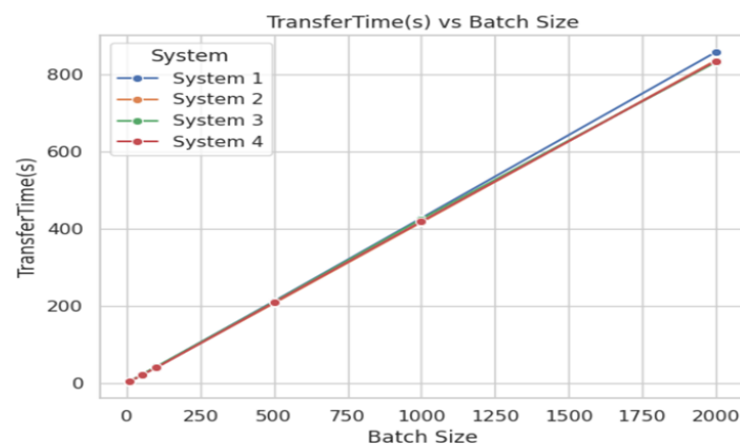


Figure 5. TransferTime (s) vs. Batch Size Across System Configurations.

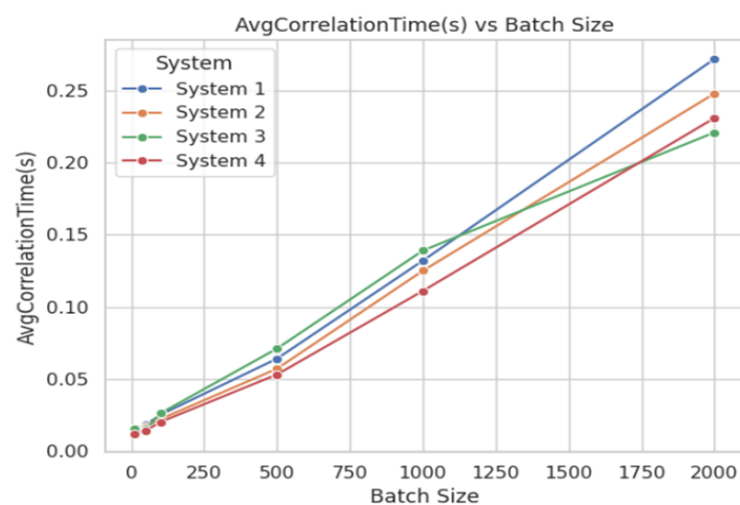


Figure 6. AvgCorrelationTime (s) vs. Batch Size Across System Configurations.

Per-Record Efficiency and Regression Analysis

As the batch size increased, the per-record times generally decreased or stabilized across all systems. Figures 7–9 present the per-record efficiency trends. Regression analysis (Figure 10) confirmed near-zero or negative slopes, indicating that the system becomes more efficient at scale.

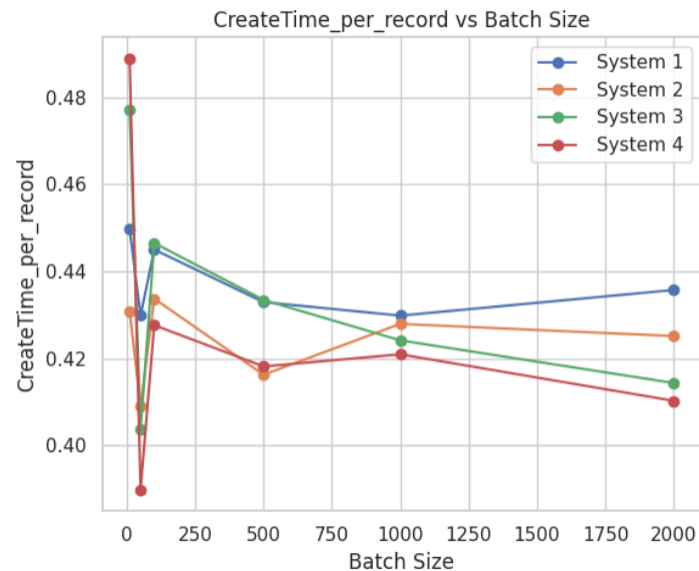


Figure 7. CreateTime per Record vs. Batch Size Across System Configurations.

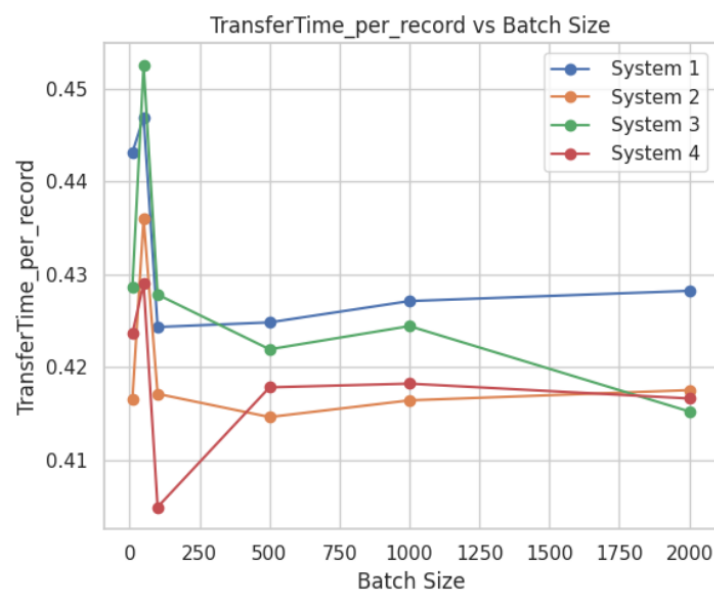


Figure 8. TransferTime per Record vs. Batch Size Across System Configurations.

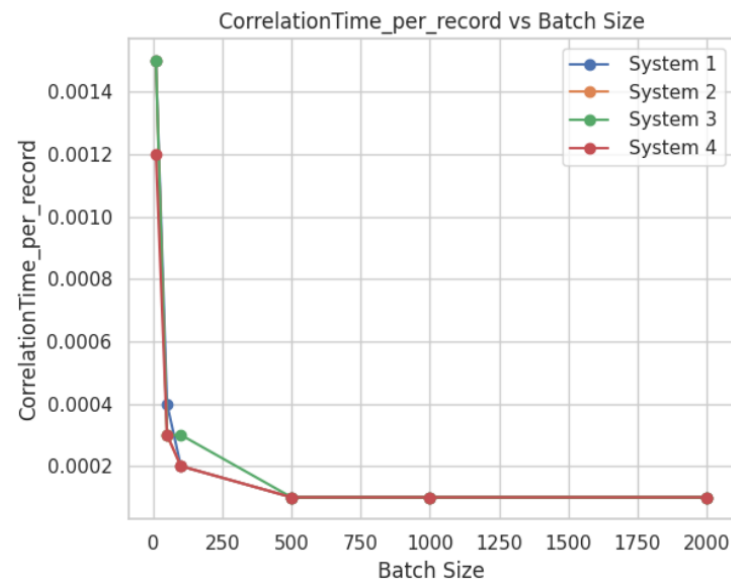


Figure 9. CorrelationTime per Record vs. Batch Size Across System Configurations.

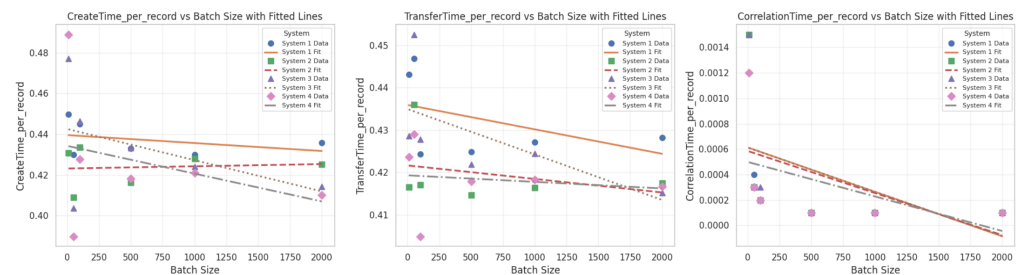


Figure 10. Linear Regression of Per-Record Operation Time vs. Batch Size Across Systems.

3.4. Concurrency Evaluation of the Blockchain Forensic System

Concurrency Results

Success rates declined as concurrency increased, with System 1 experiencing the sharpest drop beyond 1000 concurrent requests. Systems 3 and 4 showed the best resilience, as shown in Figure 11. Figures 12 and 13 further show the corresponding latency and throughput trends as concurrency levels increased across the four systems.

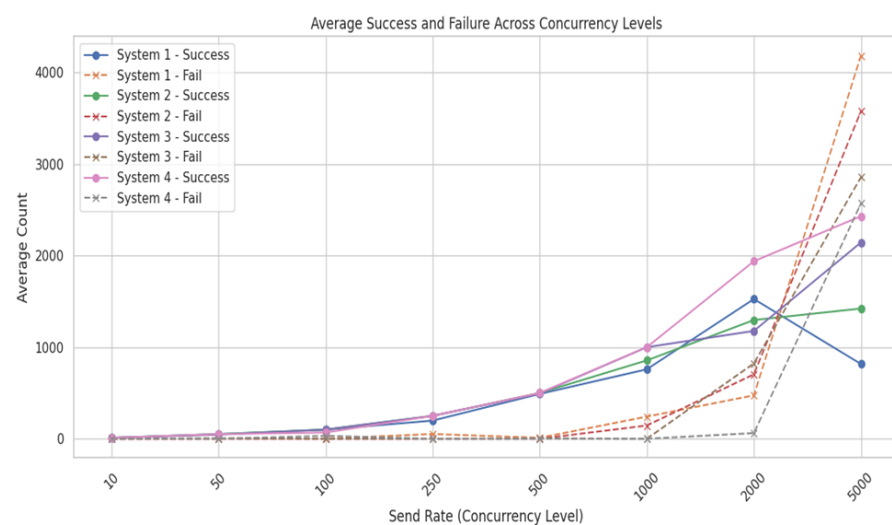


Figure 11. Average Success and Failure Counts Across Increasing Concurrency Levels.

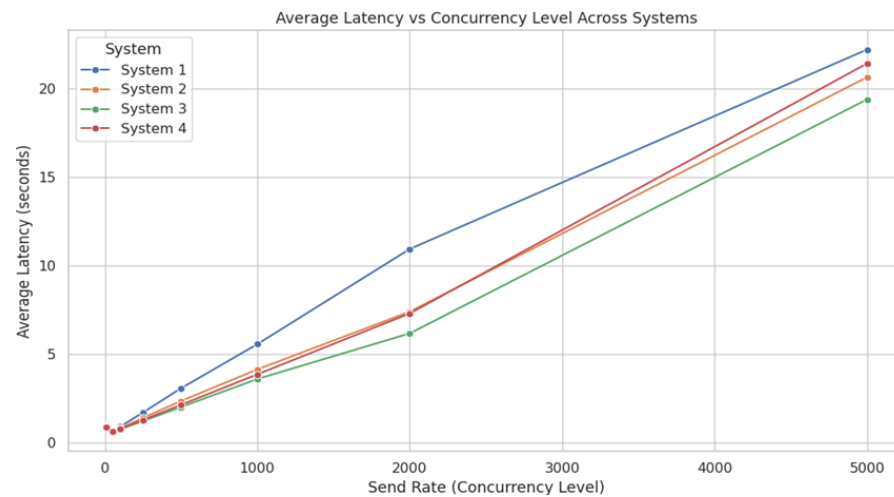


Figure 12. Average Latency vs. Concurrency Level Across Systems.

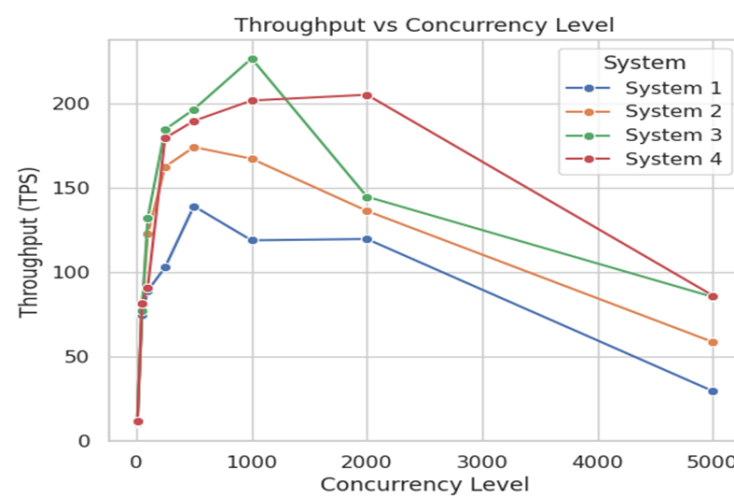


Figure 13. Throughput (TPS) vs. Concurrency Level Across Systems.

The latency results show a steady increase as concurrency levels rise, indicating predictable degradation rather than abrupt system failure. Throughput increased up to a saturation point before declining at higher concurrency levels, especially in lower-resource configurations. This pattern suggests that BlockFECS can tolerate moderate concurrent workloads, but request throttling, batching, or distributed deployment would be required for high-volume production environments. Table 8 summarises the peak throughput achieved by each system and the corresponding send rate at which the peak value occurred.

Table 8. Peak Throughput and Corresponding Concurrency Levels.

System	Peak Throughput (TPS)	Send Rate @ Peak
System 1	1500	2000
System 2	2200	2000
System 3	2400	5000
System 4	2450	5000

The results indicate that the higher-resource systems achieved their peak throughput at higher send rates, while the lower-resource systems reached saturation earlier. This supports the interpretation that additional CPU and memory capacity becomes more relevant under concurrent workloads than under small sequential transaction tests. Figure 14 provides a

regression-based view of the latency and throughput trends under increasing concurrency, further illustrating the relationship between request pressure and system performance.

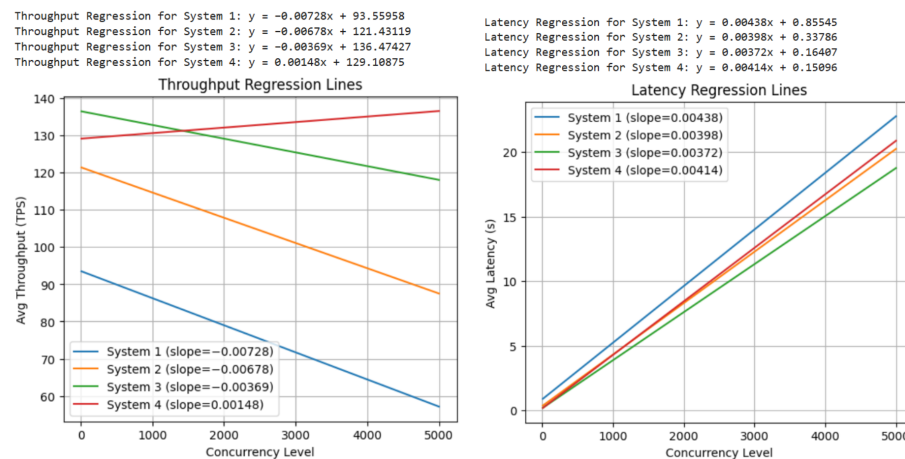


Figure 14. Linear Regression Analysis of Throughput and Latency vs. Concurrency Level.

4. Discussion

4.1. Qualitative Benchmarking and Evaluation Context

Although benchmarking against directly comparable blockchain-based forensic systems would have strengthened this study, such direct comparison was not feasible due to the emerging nature of correlation-based blockchain forensic architectures. To the best of the authors' knowledge, few published blockchain-based forensic systems combine metadata-driven evidence correlation with reported performance and scalability evaluation in a permissioned blockchain environment. Therefore, Table 9 provides a qualitative comparison with seven existing blockchain-based forensic solutions to contextualise the transaction performance observed in BlockFECS.

Table 9. Qualitative comparison of transaction performance across blockchain-based forensic systems, including BlockFECS.

Work	CPU & RAM	Duration	Transaction Time
Rathee et al. [46]	Not stated	60 s	Avg 3 s
Shen et al. [47]	AMD Ryzen 5, 16 GB	90 min	75–90 ms
Cao et al. [48]	Not stated	60 s	100–1000 ms
Hu and Li [49]	Not stated	60 min	Avg 20 s
Gul et al. [50]	X86-64, 2 GB	300 s	Create: 6.07 s
Xiao et al. [7]	Intel i7, 32 GB	60 min	Upload: 4.3 s
Lone and Mir [12]	Ubuntu, 4 GB RAM	10 rounds	Avg 11.85 s
BlockFECS	16 vCPU, 32 GB	30 trials	Create: 417 ms

The comparison shows that BlockFECS achieved lower transaction latency than several prior blockchain-based forensic systems under the reported experimental conditions. However, the comparison should be interpreted cautiously because the studies differ in blockchain platform, hardware configuration, workload duration, transaction type, and evaluation design.

4.2. System Insights and Forensic Implications

The BlockFECS system demonstrates how blockchain-backed metadata and an off-chain correlation module can support automated evidence linkage, effectively assisting in the Identification and Examination phases.

- **Feasibility:** The project demonstrates that a blockchain-backed evidence system can support semi-automated investigative assistance when computational correlation is performed off-chain rather than directly on the ledger.
- **Controlled Classification Performance:** Within the controlled proof-of-concept dataset, the algorithm achieved 1.00 precision and recall for clear Related and Duplicate categories, while Supplementary relationships remained more difficult to detect.
- **Efficient Correlation Processing:** The off-chain correlation function maintained sub-20 ms latency with throughput exceeding 60 comparisons per second across all tested systems.
- **Controlled Scalability Trends:** Regression analysis showed linear or sublinear growth within the tested workload range of up to 2000 records, indicating promising scalability behaviour under controlled experimental conditions.

Broader Implications for Enhancement of Digital Forensics

This project demonstrates that blockchain-backed infrastructures can do more than just preserve evidence. The architecture strategically separates metadata analysis from on-chain storage, reducing the performance burden while preserving the trust model.

4.3. Limitations of Study

Despite demonstrating the feasibility of combining blockchain-backed evidence management with metadata-driven correlation, this study has limitations that should guide the interpretation of the results.

- **Pairwise-Only Correlation:** The current implementation focuses on pairwise relationships between evidence items. While this supports direct linkage detection, it may overlook broader multi-evidence patterns, clusters, or event sequences that require graph-based or group-wise correlation. In addition, full pairwise comparison has $O(N^2)$ complexity when all relationships in a dataset are evaluated. Therefore, larger forensic repositories would require candidate filtering, indexing, batching, approximate similarity search, graph-based correlation, or distributed processing to reduce computational overhead.
- **Dataset Constraints:** The evaluation used an adapted dataset with synthetic forensic metadata because publicly available IoV forensic datasets with structured evidence-level metadata remain limited. As a result, the dataset supports proof-of-concept validation but does not fully represent the diversity, ambiguity, missingness, and complexity of real-world forensic evidence. In addition, the ground truth was derived from scenario logic, manual review, consistency checking, and LLM-assisted review rather than from an independently labelled real-world forensic dataset. Therefore, the reported correctness results should be interpreted as controlled proof-of-concept results rather than estimates of real-world forensic accuracy.
- **Fixed Weight Configuration:** The metadata weights used in the correlation algorithm were configured as literature-informed and scenario-specific baseline values for the selected smart city vehicular accident scenario. This study did not perform automated weight optimisation or a full sensitivity analysis to determine how different weight combinations affect classification performance. Missing or noisy metadata also reduced recall, particularly for Supplementary relationships. As a result, the current weighting scheme should not be interpreted as universally optimal across all forensic contexts. Future work should investigate sensitivity analysis, dynamic weighting, context-aware weighting, or machine-learning-based weighting strategies.
- **Prototype Scope:** BlockFECS was evaluated as a proof-of-concept system in controlled environments. Although the evaluation across four hardware configurations provides

useful evidence of controlled operational feasibility, the experiments do not represent full production-scale forensic deployment. The tests used limited workload sizes and were not conducted on a distributed multi-organisation forensic infrastructure or very large real-world evidence repositories. Therefore, the results should be interpreted as proof-of-concept evidence rather than confirmation of large-scale operational readiness. In addition, direct benchmarking against equivalent blockchain-based forensic correlation systems was limited because few existing systems report comparable metadata-correlation, scalability, and concurrency metrics under similar experimental conditions.

5. Conclusions

We proposed BlockFECS, a proof-of-concept system that extends blockchain-backed evidence management beyond secure storage and chain-of-custody logging by introducing metadata-driven evidence correlation in digital forensics. Through controlled evaluation, BlockFECS demonstrated strong performance for clear Related and Duplicate relationships, while the lower recall in the Supplementary category highlights the need for improved handling of incomplete, noisy, or ambiguous metadata. The performance, scalability-trend, and concurrency results also indicate that the system can support controlled evidence management and correlation workloads. However, further validation using larger, independently labelled, real-world forensic datasets and distributed deployment environments is required before the system can be considered ready for large-scale operational forensic use.

Future research should address these limitations through the following directions:

- Sensitivity Analysis and Dynamic Weighting: Future work should evaluate how different weight combinations affect precision, recall, and F1-score, and should investigate context-aware or machine-learning-based weighting strategies.
- Real-World Dataset Validation: Future studies should evaluate BlockFECS using larger, independently labelled forensic datasets that include natural metadata noise, missing values, inconsistent timestamps, heterogeneous devices, and ambiguous evidence relationships.
- Automatic Metadata Extraction: Future versions should integrate tools such as ExifTool and other forensic metadata extraction utilities to support direct ingestion of timestamps, device identifiers, geolocation, and file properties from real evidence sources.
- Scenario-Based and Scalable Correlation: Future work should develop correlation templates for different investigative contexts and combine them with candidate filtering, indexing, batching, approximate similarity search, graph-based correlation, or distributed processing to reduce computational overhead as evidence repositories grow.
- Integration with Intelligent IoT Analysis: Future work could integrate anomaly detection or machine-learning-based IoT analysis methods to support candidate filtering, dynamic weighting, and more complex relationship discovery while preserving explainability and forensic auditability.

To conclude, BlockFECS demonstrates the controlled feasibility of using blockchain-backed infrastructure as a foundation for trustworthy and metadata-driven forensic assistance. While the current implementation remains a proof-of-concept, it provides a practical foundation for future research on scalable, explainable, and audit-ready evidence correlation within digital forensic evidence management systems.

Author Contributions: Conceptualization, O.S.I.; methodology, O.S.I., M.B.A. and S.G.; software, O.S.I.; validation, O.S.I., M.B.A. and S.G.; formal analysis, O.S.I.; investigation, O.S.I.; data curation, O.S.I.; writing—original draft preparation, O.S.I.; writing—review and editing, O.S.I., M.B.A. and S.G.; visualization, O.S.I.; supervision, M.B.A. and S.G.; project administration, O.S.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in this study are openly available in GitHub at <https://github.com/igsamoshoke/BLOCKFECS-POC-DATASET> (accessed on 11 June 2026). The repository contains the proof-of-concept forensic metadata datasets, correlation evaluation data, stress-test data, and experimental outputs used to support the findings reported in this study.

Acknowledgments: The authors acknowledge the support and research environment provided by the School of Information and Communication Technology, University of Tasmania.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Smart Contracts Pseudo-Codes

Algorithm A1 CreateEvidence Transaction

Require: *evidenceID, title, description, timestamp, latitude, longitude, deviceId, userId, caselId, fileType, fileHash, googleDriveId*

Ensure: Stores the evidence on blockchain and returns success message

```

1: if Caller's MSP ≠ Org1MSP then
2:   Deny access and exit
3: end if
4: if Any required field is empty then
5:   Return error: missing fields
6: end if
7: if Evidence already exists in ledger then
8:   Return error: duplicate evidence
9: end if
10: Create new evidence object with all metadata fields
11: Initialize TransferHistory with current owner
12: Initialize TimestampHistory with current time
13: Store evidence object using evidenceID as key
14: Log the action as Create in blockchain logs
15: return "SUCCESS"

```

Algorithm A2 TransferEvidence Transaction

Require: *evidenceID, newOwner*

Ensure: Updates ownership and transfer history

```

1: if Caller's MSP ∉ {Org1MSP, Org2MSP} then
2:   Deny access and exit
3: end if
4: Retrieve evidence object by evidenceID
5: if Evidence not found then
6:   Return error
7: end if
8: Update Owner field to newOwner
9: Append newOwner to TransferHistory
10: Append current time to TimestampHistory
11: Save updated evidence object
12: Log the action as Transfer in blockchain logs

```

Algorithm A3 DeleteEvidence Transaction

Require: *evidenceID***Ensure:** Marks evidence as deleted in ledger

- 1: **if** Caller's MSP $\notin$ {Org1MSP, Org2MSP} **then**
 - 2: Deny access and exit
 - 3: **end if**
 - 4: Retrieve evidence object by *evidenceID*
 - 5: **if** Evidence not found **then**
 - 6: Return error
 - 7: **end if**
 - 8: Set Owner field to "DELETED"
 - 9: Save updated evidence object
 - 10: Log the action as Delete in blockchain logs
-

Algorithm A4 EditEvidence Transaction

Require: *evidenceID, newTitle, newDesc***Ensure:** Updates title and description of evidence

- 1: **if** Caller's MSP $\notin$ {Org1MSP, Org2MSP} **then**
 - 2: Deny access and exit
 - 3: **end if**
 - 4: Retrieve evidence object by *evidenceID*
 - 5: **if** Evidence not found **then**
 - 6: Return error
 - 7: **end if**
 - 8: Update EvidenceTitle and Description
 - 9: Save updated evidence object
 - 10: Log action as Edit with old and new values
-

Algorithm A5 GetEvidenceDetails Function

Require: *evidenceID***Ensure:** Returns details of specified evidence

- 1: Retrieve value from ledger using *evidenceID* as key
 - 2: **if** value is nil **then**
 - 3: Return error: evidence not found
 - 4: **end if**
 - 5: Deserialize value into evidence object
 - 6: **return** evidence object
-

Algorithm A6 QueryEvidenceByCaseID Function

Require: *caseId***Ensure:** Returns all evidence items associated with the specified case ID

- 1: Query all ledger entries using `GetStateByRange("", "")`
 - 2: Initialize *results* as an empty list
 - 3: **for all** key-value pair in result **do**
 - 4: Attempt to deserialize value into an evidence object
 - 5: **if** deserialization fails **then**
 - 6: Skip this record
 - 7: **end if**
 - 8: **if** evidence.CaseID equals *caseId* **then**
 - 9: Append evidence object to *results*
 - 10: **end if**
 - 11: **end for**
 - 12: **return** *results*
-

Algorithm A7 GetAllEvidence Function**Require:** None**Ensure:** Returns a list of all evidence items stored on the blockchain

```

1: Query all ledger entries using GetStateByRange("", "")
2: for all key-value pair in result do
3:   if key starts with "LOG-" then
4:     Skip entry (not evidence)
5:   end if
6:   Deserialize value into evidence object
7:   Append to results list
8: end for
9: return results list

```

Algorithm A8 GetTransactionLogs Function**Require:** None**Ensure:** Chronologically ordered list of transaction logs

```

1: Query all ledger entries using GetStateByRange("", "")
2: for all key-value pair in result do
3:   if key does not start with "LOG-" then
4:     Skip entry (not a log)
5:   end if
6:   Deserialize value into log object
7:   Append to log list
8: end for
9: Sort log list by Timestamp field (ascending)
10: return sorted log list

```

References

1. Atlam, H.F.; Hemdan, E.E.D.; Alenezi, A.; Alassafi, M.O.; Wills, G.B. Internet of things forensics: A review. *Internet Things* **2020**, *11*, 100220. [\[CrossRef\]](#)
2. Miao, J.; Wang, Z.; Wu, Z.; Ning, X.; Tiwari, P. A blockchain-enabled privacy-preserving authentication management protocol for Internet of Medical Things. *Expert Syst. Appl.* **2024**, *237*, 121329.
3. Saigopal, V.V.R.G.; Raju, V. IIOT digital forensics and major security issues. In Proceedings of the International Conference on Computational Intelligence (ICCI), Pune, India, 12–13 December 2020; pp. 233–236.
4. Cebe, M.; Erdin, E.; Akkaya, K.; Aksu, H.; Uluagac, S. Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Commun. Mag.* **2018**, *56*, 50–57. [\[CrossRef\]](#)
5. Igonor, O.S.; Amin, M.B.; Garg, S. The Application of Blockchain Technology in the Field of Digital Forensics: A Literature Review. *Blockchains* **2025**, *3*, 5. [\[CrossRef\]](#)
6. Mahrous, W.A.; Farouk, M.; Darwish, S.M. An enhanced blockchain-based IoT digital forensics architecture using fuzzy hash. *IEEE Access* **2021**, *9*, 151327–151336.
7. Xiao, N.; Wang, Z.; Sun, X.; Miao, J. A novel blockchain-based digital forensics framework for preserving evidence and enabling investigation in industrial Internet of Things. *Alex. Eng. J.* **2024**, *86*, 631–643.
8. Chopade, M.; Khan, S.; Shaikh, U.; Pawar, R. Digital forensics: Maintaining chain of custody using blockchain. In Proceedings of the Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 744–747.
9. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–34. [\[CrossRef\]](#)
10. Khan, S.N.; Loukil, F.; Ghedira-Guegan, C.; Benkhelifa, E.; Bani-Hani, A. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-Peer Netw. Appl.* **2021**, *14*, 2901–2925. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Stephen, R.; Alex, A. A review on blockchain security. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2018; Volume 396, p. 012030.
12. Lone, A.H.; Mir, R.N. Forensic-chain: Blockchain based digital forensics chain of custody with PoC in Hyperledger Composer. *Digit. Investig.* **2019**, *28*, 44–55.

13. Tyagi, R.; Sharma, S.; Mohan, S. Blockchain enabled intelligent digital forensics system for autonomous connected vehicles. In Proceedings of the International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, 10–11 March 2022; pp. 1–6.
14. Yan, W.; Shen, J.; Cao, Z.; Dong, X. Blockchain based digital evidence chain of custody. In Proceedings of the 2nd International Conference on Blockchain Technology, Hilo, HI, USA, 12–14 March 2020; pp. 19–23.
15. Liu, G.; He, J.; Xuan, X. A data preservation method based on blockchain and multidimensional hash for digital forensics. *Complexity* **2021**, 2021, 5536326. [\[CrossRef\]](#)
16. Li, M.; Lal, C.; Conti, M.; Hu, D. LEChain: A blockchain-based lawful evidence management scheme for digital forensics. *Future Gener. Comput. Syst.* **2021**, 115, 406–420. [\[CrossRef\]](#)
17. Akbarfam, A.J.; Dorai, G.; Maleki, H. Secure Cross-Chain Provenance for Digital Forensics Collaboration. In Proceedings of the IEEE 6th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA), Washington, DC, USA, 28–31 October 2024; pp. 389–398.
18. Nyalety, E.; Parizi, R.M.; Zhang, Q.; Choo, K.K.R. BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability. In Proceedings of the IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 18–25.
19. Chen, S.; Zhao, C.; Huang, L.; Yuan, J.; Liu, M. Study and implementation on the application of blockchain in electronic evidence generation. *Forensic Sci. Int. Digit. Investig.* **2020**, 35, 301001. [\[CrossRef\]](#)
20. Philip, A.O.; Saravanaguru, R.K. Smart contract based digital evidence management framework over blockchain for vehicle accident investigation in IoV era. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, 34, 4031–4046. [\[CrossRef\]](#)
21. Khan, A.A.; Shaikh, A.A.; Laghari, A.A. IoT with multimedia investigation: A secure process of digital forensics chain-of-custody using blockchain hyperledger sawtooth. *Arab. J. Sci. Eng.* **2023**, 48, 10173–10188.
22. Ragu, G.; Ramamoorthy, S. A blockchain-based cloud forensics architecture for privacy leakage prediction with cloud. *Healthc. Anal.* **2023**, 4, 100220. [\[CrossRef\]](#)
23. Pourvahab, M.; Ekbatanifard, G. An efficient forensics architecture in software-defined networking-IoT using blockchain technology. *IEEE Access* **2019**, 7, 99573–99588.
24. Hu, S.; Zhang, S.; Fu, K. Tfcchain: Blockchain-based trusted forensics scheme for mobile phone data whole process. In Proceedings of the IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 4–6 March 2022; Volume 6, pp. 155–165.
25. Rane, S.; Dixit, A. BlockSLaaS: Blockchain Assisted Secure Logging-as-a-Service for Cloud Forensics. In *Proceedings of the International Conference on Security & Privacy*; Springer: Singapore, 2019; pp. 77–88.
26. Bhattacharjee, T.; Mahapatra, A.S.; De, D.; Chowdhury, A. JudicBlock: Judicial Evidence Preservation Scheme Based on Blockchain Technology. *Blockchains* **2025**, 3, 11. [\[CrossRef\]](#)
27. Mbimbi, B.; Murray, D.; Wilson, M. Private Blockchain-Driven Digital Evidence Management Systems: A Collaborative Mining and NFT-Based Framework. *Information* **2025**, 16, 616. [\[CrossRef\]](#)
28. Mahajan, A.; Manchanda, B.; Aggarwal, H.; Mandhana, H.; Bhatt, K.; Sharma, N.; Kaur, J.; Singh, A. ChainSEAL: A Blockchain-Based Secure and Adaptive Evidence Ledger. In Proceedings of the 18th International Conference on COMMunication Systems and NETworks (COMSNETS), Bengaluru, India, 6–10 January 2026; pp. 1185–1189. [\[CrossRef\]](#)
29. García Fernández, D.; Tobarra, L.; Robles-Gómez, A.; Pastor Vargas, R. Agentless and Automated Acquisition of Digital Evidence in Corporate and Industrial Networks: Architecture, Validation, and Compliance. *Electronics* **2026**, 15, 744. [\[CrossRef\]](#)
30. Haji, I.A.; Mohammed, S.D.; Mousa, K.M. Blockchain Based Chain of Custody and Digital Evidence Legality in Post Conflict Prosecutions. *Front. Blockchain* **2026**, 9, 1801364. [\[CrossRef\]](#)
31. Nedaković, A.; Hasselgren, A.; Kralevska, K.; Gligoroski, D. Hyperledger fabric platform for healthcare trust relations—Proof-of-Concept. *Blockchain Res. Appl.* **2023**, 4, 100156. [\[CrossRef\]](#)
32. Cachin, C. Architecture of the hyperledger blockchain fabric. In Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers, Chicago, IL, USA, 25 July 2016; Volume 310, pp. 1–4.
33. Hyperledger Project. Hyperledger Composer (README) [Computer Software]. 2019. Available online: <https://github.com/hyperledger-archives/composer> (accessed on 12 December 2025).
34. Akinbi, A.; MacDermott, A.; Ismael, A.M. A systematic literature review of blockchain-based Internet of Things (IoT) forensic investigation process models. *Forensic Sci. Int. Digit. Investig.* **2022**, 42, 301470.
35. Atlam, H.F.; Ekuri, N.; Azad, M.A.; Lallie, H.S. Blockchain Forensics: A Systematic Literature Review of Techniques, Applications, Challenges, and Future Directions. *Electronics* **2024**, 13, 3568. [\[CrossRef\]](#)
36. Khanji, S.; Alfandi, O.; Ahmad, L.; Kakkengal, L.; Al-Kfairy, M. A systematic analysis on the readiness of Blockchain integration in IoT forensics. *Forensic Sci. Int. Digit. Investig.* **2022**, 42, 301472.
37. Schatz, B.; Mohay, G.; Clark, A. A correlation method for establishing provenance of timestamps in digital evidence. *Digit. Investig.* **2006**, 3, 98–107. [\[CrossRef\]](#)

38. Casey, E.; Turvey, B.E. *Investigative Reconstruction with Digital Evidence*; Academic Press: London, UK, 2011.
39. Jiang, B.; Chen, M.; Chen, F. A clock drift compensation method for synchronous sampling in sensor networks. *Meas. Sci. Technol.* **2019**, *30*, 025103. [[CrossRef](#)]
40. Casey, E.; Jaquet-Chiffelle, D.O.; Spichiger, H.; Ryser, E.; Souvignet, T. Structuring the evaluation of location-related mobile device evidence. *Forensic Sci. Int. Digit. Investig.* **2020**, *32*, 300928. [[CrossRef](#)]
41. Matte, C.; Achara, J.P.; Cunche, M. Device-to-identity linking attack using targeted wi-fi geolocation spoofing. In Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks, New York, NY, USA, 22–26 June 2015; pp. 1–6.
42. Rak, R.; Kopencova, D.; Felcan, M. Digital vehicle identity—Digital VIN in forensic and technical practice. *Forensic Sci. Int. Digit. Investig.* **2021**, *39*, 301307.
43. Nguyen, T.D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; Sadeghi, A.R. D²IoT: A Federated Self-learning Anomaly Detection System for IoT. In Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 756–767. [[CrossRef](#)]
44. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Comput.* **2018**, *17*, 12–22. [[CrossRef](#)]
45. Moustafa, D. Ton_iiot Datasets for Cybersecurity Applications Based Artificial Intelligence. 2019. Available online: <https://research.unsw.edu.au/projects/unsw-nb15-dataset> (accessed on 12 December 2025).
46. Rathee, G.; Ahmad, F.; Jaglan, N.; Konstantinou, C. A secure and trusted mechanism for industrial IoT network using blockchain. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1894–1902. [[CrossRef](#)]
47. Shen, M.; Liu, H.; Zhu, L.; Xu, K.; Yu, H.; Du, X.; Guizani, M. Blockchain-assisted secure device authentication for cross-domain industrial IoT. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 942–954.
48. Cao, Y.; Jia, F.; Manogaran, G. Efficient traceability systems of steel products using blockchain-based industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2019**, *16*, 6004–6012. [[CrossRef](#)]
49. Hu, W.; Li, H. A blockchain-based secure transaction model for distributed energy in Industrial Internet of Things. *Alex. Eng. J.* **2021**, *60*, 491–500. [[CrossRef](#)]
50. Gul, M.J.; Subramanian, B.; Paul, A.; Kim, J. Blockchain for public health care in smart society. *Microprocess. Microsyst.* **2021**, *80*, 103524. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.