



LEGAL DOCUMENT SUMMARIZATION AND INTEGRITY VERIFICATION FRAMEWORK

Using Hybrid NLP and Consortium Blockchain

¹Sumati Gaur, ²Shruti Agarwal, ³Durga Shankar Shukla

¹Research Scholar, ²Research Scholar, ³Assistant Professor

^{1,2,3}School of Computer Applications (SOCA)

^{1,2,3}Babu Banarasi Das University, Lucknow, Uttar Pradesh, India

¹sumati.gaur1007@gmail.com, ²shruti.agarwal0112@gmail.com, ³mr.dsshukla@bbdu.ac.in

Abstract : Courts and police departments still rely on paper records. These legal documents are large, disorganized and difficult to analyze. The issue is particularly relevant when a dormant case is formally reopened, since the investigator needs to analyze a document from 2010 and compare it with findings from 2026 in order to find what claims are still valid, what alibis are valid and what claims were never considered [1, 2]. This paper provides an approach which combines domain-adapted natural language processing (NLP) and enterprise consortium blockchain to achieve automated evidence summarization, temporal delta analysis and tamper-evident document retrieval [3]. First, an off-chain NLP engine anonymizes personal information in the text with Legal Named Entity Recognition (LegNER) algorithm according to GDPR and India's DPDP Act [4]. Then, an extractive RoBERTa and abstractive Legal-LED summarizers generate summary of each document in Subject-Predicate-Object (S-P-O) keypoint format [2, 3]. Cosine similarity compares testimony against crime-scene logs, and a temporal delta engine sorts the facts in a reopened file into confirmed, contradicted, new and unaddressed marking. A storage and ledger engine encrypts the raw files for off-chain InterPlanetary File System (IPFS) storage and records SHA-256 digests and keypoint indices on a Hyperledger Fabric ledger [5, 6]. Keeping heavy neural processing off-chain and writing only hashes and keypoint indices to the ledger means that any later change to a record can be detected.

Keywords - Legal Document Summarization, Natural Language Processing (NLP), Permissioned Blockchain, Hyperledger Fabric, Temporal Delta Analysis, Subject-Predicate-Object (S-P-O) Triples, IPFS, SHA-256 Hashing, LegNER, Scene Matching, Cold Case Analytics

I. INTRODUCTION

A. Background and Context

Many legal systems are struggling with case backlogs, scattered paper records and slow evidence checks [1, 7]. Judges, prosecutors, defense lawyers and police investigators routinely go through hundreds of pages per case, from police logs and witness statements to forensic disclosures and statutory filings [1]. Things get harder when a dormant cold case is reopened after decades. Suppose set a case file written in 2010 beside forensic reports produced in 2026 then someone has to read them line by line. They are looking for newly uncovered facts, alibis that no longer hold, and old claims that nobody ever addressed.

Natural language processing (NLP) has become a practical tool for legal text [1, 8, 9]. Transformer-based models such as Legal-BERT, Legal-LED and instruction-tuned large language models (LLMs) can recognize named entities, classify contract clauses and summarize court judgments [2, 4, 8]. However, using them directly into a legal workflow brings two risks. The first is hallucination. Generative models sometimes create its own content or flip the meaning, for instance dropping a negation such as "not," is enough to reverse the legal meaning of a judicial summary [1, 8]. The second is fragile, centralized storage. Conventional databases offer no cryptographic immutability, so digitized evidence and AI-generated outputs can be edited without trace, corrupted, or separated from their chain of custody [10].

We propose to resolve these issues with a dual-engine architecture [3]. An off-chain deep learning pipeline that handles entity masking, hybrid summarization, S-P-O keypoint extraction, scene matching and temporal delta tracking. An on-chain consortium blockchain layer then anchors SHA-256 hashes and holds a searchable keypoint index, enabling nobody to dig through paper case files anymore [6].

B. Core Research Contributions

We make five contributions -

- 1) **A dual-engine architecture (NLP plus blockchain):** Deep transformer models run off-chain, while an enterprise Hyperledger Fabric consortium ledger handles verification on-chain. The split gives zero-trust integrity without sacrificing throughput [6, 11].
- 2) **Entity extraction and anonymization:** A LegNER model masks personally identifiable information (PII) before anything is stored or processed further, supporting GDPR Article 17 and DPDP compliance [4].
- 3) **Hybrid S-P-O summarization:** RoBERTa extractive filtering followed by Legal-LED abstractive generation turns each document into S-P-O factual keypoint triples, which keeps the abstractive stage from drifting away from the source [2, 3].
- 4) **Longitudinal temporal delta analytics (2010→2026):** A temporal graph-alignment decision engine separates confirmed, contradicted and newly discovered evidence when a historical cold case is reopened.
- 5) **Paperless indexing and tamper-evident retrieval:** S-P-O keypoints are indexed in smart contracts (chaincode) alongside IPFS Content Identifiers (CIDs), so precedents can be found with Jaccard keypoint queries instead of a manual paper search [12].

Section II reviews related work, Section III describes the architecture, and Section IV gives the mathematical formulation. Section V covers the blockchain layer, Section VI deals with legal and ethical questions, and Section VII concludes.

II. LITERATURE REVIEW

A. Legal NLP and Hybrid Summarization

Early work on legal text summarization relied on extractive graph algorithms such as LexRank and TextRank [1]. Extraction preserves the exact statutory wording, but the summaries it produces tend to be disjointed and hard to follow [1]. Pre-trained transformers made abstractive summarization possible, and benchmark corpora such as IN-Abs, IN-Ext and UK-Abs gave the field common ground for evaluation [2]. Shukla et al. [2] showed that long-context models like Legal-LED handle extended judicial arguments well, although abstractive models remain susceptible to factual hallucination when nothing constrains them.

Recent systems try to contain that risk with two-stage designs. LeSum [3] and HybridSumm [13] first isolate the critical sentences with an extractive stage and then pass them to an abstractive generator that condenses the text without altering key legal facts. Sadikot and Jain [14] took a different route with SemA, a zero-shot agentic workflow. They show that modular prompt orchestration across coordinated LLM agents can reach high semantic fidelity without intensive domain fine-tuning.

B. Blockchain-Based Legal Provenance and Metadata Security

Permissioned blockchains such as Hyperledger Fabric offer an alternative to public networks. They bring identity-based membership and access control, careful handling of non-deterministic code, and no dependence on a native cryptocurrency, so there are no gas fees [6]. Ahmed et al. [15] showed that NLP models can pull contractual obligations out of raw legal text and map them into self-executing smart contracts.

Storing full text or large binary files on-chain is a poor idea, though. It bloats the ledger state and conflicts with data protection laws such as the GDPR and the DPDP Act [16]. Off-chain storage, for example on IPFS, avoids both problems: encrypted binaries stay off the ledger, and only lightweight CIDs and SHA-256 hashes are anchored on-chain [5, 11, 16].

Table 1 places the closest related studies side by side. Each of them covers part of the problem, and the last column states what the present work adds.

Table 1. Summary of related work and the gap addressed in this study.

Author(s) and year	Domain scope	Methodology / models	Key benchmark / result	Gap addressed in this work
Kanapala et al. (2021) [1]	Legal NLP survey	Extractive vs. abstractive review	Identifies extreme text length and jargon as barriers	Establishes the need for hybrid, domain-specific pipelines
Shukla et al. (2022) [2]	Judicial opinions	Legal-BERT, Legal-LED, BART, PEGASUS	Benchmarks on IN-Abs, IN-Ext and UK-Abs	Addresses abstractive hallucination in pure generation models
Ahmed et al. (2024) [15]	Smart contracts	Legal entity extraction and blockchain	Automates contract logic generation	Extends the focus from contracts to litigation case differences
Sadikot and Jain (2025) [14]	Zero-shot agents	Modular Agentic Workflow (SemA)	CivilSum BERTScore F1: 0.8902	Removes external API dependency through local hybrid models
Ansar et al., LeSum (2026) [3]	Cost-effective summarization	Extractive filter and abstractive LLM	Context reduction and high factual recall	Adds an integrated cryptographic integrity layer
Govea et al. (2026) [8]	Contract automation	Domain encoders vs. instruction-tuned LLMs	Clause classification and extraction benchmarking	Extends processing to full court case proceedings
Saxena (2025) [17]	Judicial assistance review	Extractive and transformer summarizers (Legal-BERT, GPT-based)	ROUGE, BLEU and human ratings; reports 35–40% faster case review	Leaves integrity and auditability of generated summaries to future work

III. END-TO-END SYSTEM ARCHITECTURE

The system keeps heavy neural text processing away from ledger updates by splitting the work between an off-chain NLP engine and an on-chain blockchain engine [11].

A. Global System Architecture Pipeline

Fig. 1 shows the overall flow, from raw document ingestion through off-chain NLP analysis and encrypted IPFS storage to smart-contract indexing on the ledger. The pipeline has four tiers.

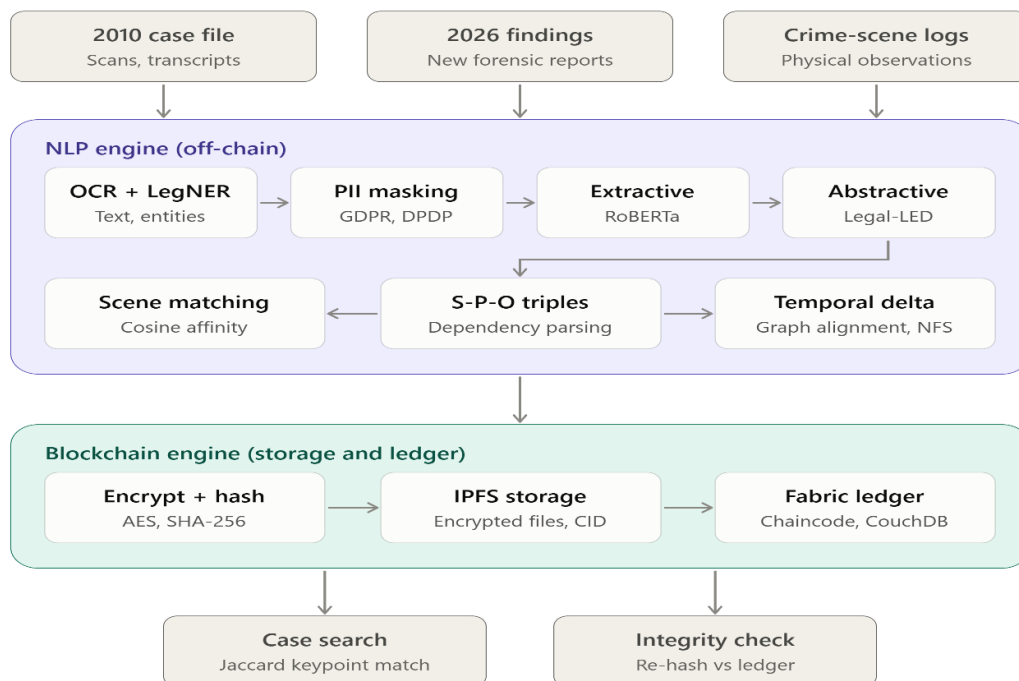


Fig. 1. Global system architecture: an off-chain NLP engine feeds an on-chain blockchain engine.

- **Multi-source ingestion tier** - This tier takes in historical court records (scanned PDFs and 2010 transcripts), the reopened 2026 findings (forensic reports) and physical crime-scene logs.
- **Off-chain NLP engine** - Six steps run in sequence:
 - *OCR and LegNER* convert raw image scans into structured text and tag legal entities (JUDGE, LAW, PERSON, CASE_REF) [4].
 - *PII masking* redacts personal identification details to stay within GDPR Article 17 and India's DPDP Act [4].
 - *Extractive filtering (RoBERTa)* scores sentence salience, strips non-informational boilerplate and shortens the text by 60% [3].
 - *Abstractive generation (Legal-LED)* writes narrative summaries using pointer-generator networks, which preserve rare legal terms [2].
 - *S-P-O parsing* extracts canonical Subject-Predicate-Object keypoint triples through dependency parsing.
 - *Scene matching and temporal delta* compute cosine embedding affinity against crime-scene logs and run graph alignment to track factual differences between 2010 and 2026.
- **Blockchain engine (storage and ledger)** - Files are encrypted with AES-256, SHA-256 digests are generated, encrypted binaries go to IPFS, and the resulting CIDs and keypoint indices are committed to a Hyperledger Fabric ledger [5, 6].
- **User verification tier** - Researchers can search precedents without paper by using Jaccard keypoint matching and can check integrity instantly by comparing a local re-hash of a file against the ledger [18].

B. Decoupled Software Implementation Architecture

The prototype is built in layers and runs either with a cloud connection or fully offline (Fig. 2). There are five.

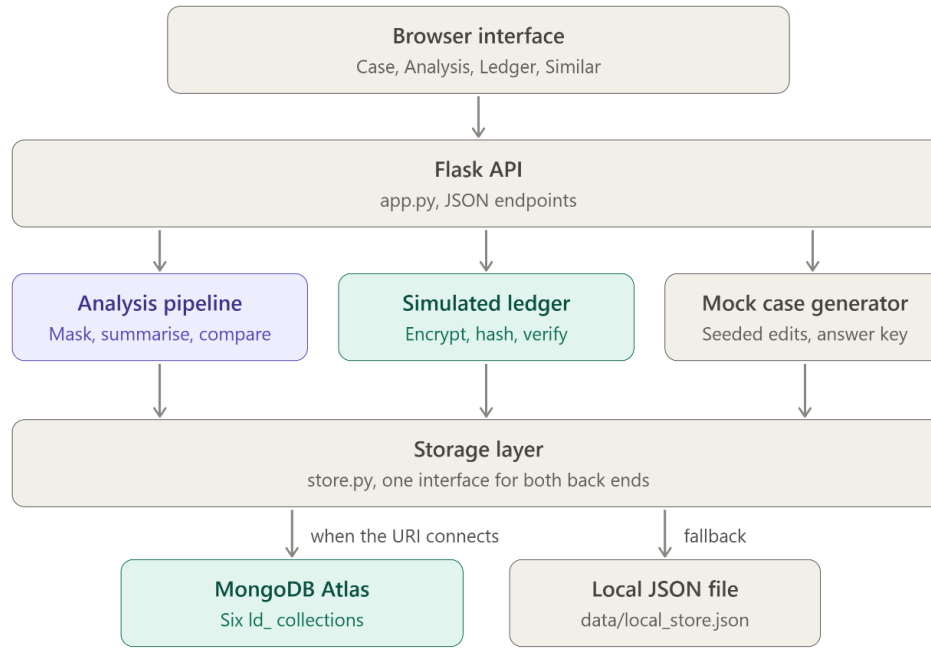


Fig. 2. Software implementation architecture showing the decoupled system layers.

- **Browser interface:** A web dashboard with four tabs: Case Ingestion, Temporal Analysis, Ledger Verification and Similar Case Discovery.
- **Flask API core (app.py):** RESTful JSON endpoints that orchestrate NLP pipeline calls, ledger hashing and search queries.
- **Core execution modules:** The *analysis pipeline* performs entity masking, hybrid summarization, S-P-O triple generation and temporal diffing. The *simulated ledger* carries out AES encryption, SHA-256 hashing and mock chaincode operations. The *mock case generator* injects seeded text edits and adversarial modifications so that tamper detection can be tested.
- **Unified storage interface (store.py):** An abstraction layer between the application logic and data persistence.
- **Dual persistence backend:** The system connects to a cloud-hosted MongoDB Atlas instance (six specialized ld_collections) when online and falls back automatically to a local JSON repository (data/local_store.json) when offline.

IV. MATHEMATICAL FORMULATIONS AND TEMPORAL DELTA LOGIC

A. S-P-O Extraction and Crime Scene Affinity Scoring

We represent both legal transcripts and physical crime-scene logs as sets of Subject-Predicate-Object (S-P-O) factual triples:

$$\mathcal{K}_{\chi_{\alpha\sigma\epsilon}} = \{(s_i, p_i, o_i)\}_{i=1}^N, \quad \mathcal{K}_{\sigma\chi_{\epsilon\vee\epsilon}} = \{(s'_j, p'_j, o'_j)\}_{j=1}^M \quad (1)$$

Each triple is mapped into a vector space by a fine-tuned legal sentence transformer E_ϕ . The match affinity score $S(\mathcal{K}_{\text{case}}, \mathcal{K}_{\text{scene}})$ is the average, over all triples in the case file, of each triple's best cosine similarity to any triple in the scene log (max-pooled cosine similarity):

$$S(\mathcal{K}_{\chi_{\alpha\sigma\epsilon}}, \mathcal{K}_{\sigma\chi_{\epsilon\vee\epsilon}}) = \frac{1}{|\mathcal{K}_{\chi_{\alpha\sigma\epsilon}}|} \sum_{u \in \mathcal{K}_{\chi_{\alpha\sigma\epsilon}}} \max_{v \in \mathcal{K}_{\sigma\chi_{\epsilon\vee\epsilon}}} \left(\frac{E_\phi(u) \cdot E_\phi(v)}{\|E_\phi(u)\| \|E_\phi(v)\|} \right) \quad (2)$$

If $S(\mathcal{K}_{\text{case}}, \mathcal{K}_{\text{scene}}) \geq \tau_{\text{match}}$, with $\tau_{\text{match}} = 0.75$, the testimony is treated as consistent with the crime-scene log. Otherwise the system raises an evidence-mismatch alert.

B. Temporal Delta Decision Logic ($\Delta_{2010 \rightarrow 2026}$)

When a historical case file ($t_1 = 2010$) is reopened alongside modern findings ($t_2 = 2026$), the temporal delta engine places every newly introduced fact into one of four mutually exclusive states: New, Contradicted, Confirmed or Unaddressed. Fig. 3 shows the decision flow applied to each new fact.

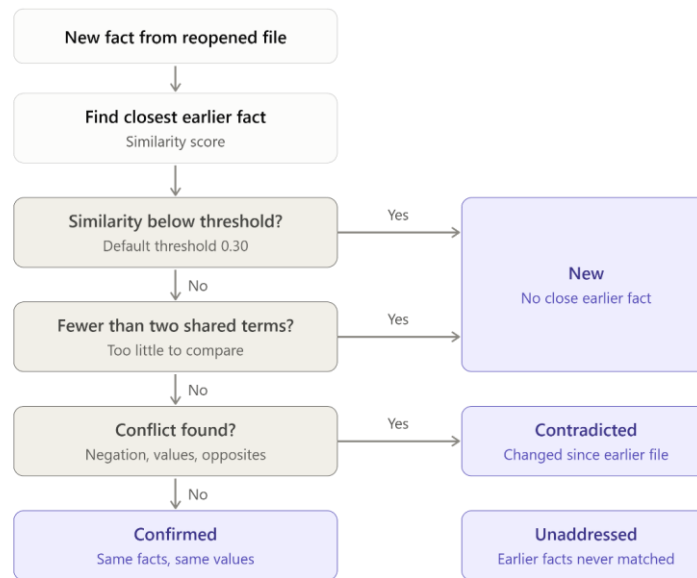


Fig. 3. Temporal delta decision flowchart for the 2010→2026 temporal analysis.

The classification works through four checks in order.

- 1) **Semantic similarity check** - The candidate fact f_{2026} is compared against all historical facts f_{2010} using cosine embedding distance. If the highest similarity S_{sem} is below 0.30, the fact is classed as New.
- 2) **Shared-term check** - If $S_{\text{sem}} \geq 0.30$, the pair is tested for term overlap. Fewer than two shared key legal terms mean there is too little context to compare, so the fact is again classed as New.
- 3) **Conflict detection** - The pair is screened for semantic contradiction, such as negation modifiers, conflicting numeric values or opposite legal antonyms. A conflict makes the fact Contradicted, for instance when a historical alibi is disproven by DNA re-testing in 2026.
- 4) **Confirmation** - If there is no conflict, the fact is Confirmed: same facts, same values. Historical facts in f_{2010} that no 2026 entry ever matched are marked Unaddressed.

The Novel Finding Score (NFS) quantifies whether a 2026 statement $v_j(2026)$ is an unaddressed discovery. Let D_{ij} be the cosine distance between a 2010 fact and a 2026 fact; then

$$D_{ij} = 1 - \frac{v_i(2010) \cdot v_j(2026)}{\|v_i(2010)\| \|v_j(2026)\|}, \quad \text{NFS} \left(v_j(2026) \right) = \min_i D_{ij} \quad (3)$$

A high score ($\text{NFS} > 0.70$) marks $v_j(2026)$ as a newly uncovered finding that should be folded into the updated summary. Because $\text{NFS} = 1 - S_{\text{sem}}$, this threshold is the same condition as the first check above.

V. BLOCKCHAIN STORAGE, INDEXING, AND SECURITY LAYER

A. Hybrid Storage Architecture and Privacy Guarantees

To keep the ledger lean and stay within data protection rules (GDPR Article 17 and the DPDP Act), we store raw evidence files and complete summaries off-chain [11, 16]. Payloads are encrypted with AES-256 before they go to IPFS, which returns a 46-character CID [5, 19]. The Hyperledger Fabric state store on-chain keeps only a small verification record, shown with illustrative values in Listing 1. It holds hashes, an off-chain pointer and masked keypoints, not the case text itself.

Listing 1. Example on-chain verification record (illustrative values).

```

{
  "case_id": "CASE-2010-2026-0892",
  "case_name": "State vs. Appellant_Redacted",
  "doc_sha256": "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
  "summary_sha256": "8f434346648f6b96df89dda901c5176b10a6d83961dd3c1ac88b59b2dc327aa4",
  "ipfs_cid": "QmXoypi2jW3WknFiJnKLwHCnL72vedxjQkDDPlmXWo6uco",
  "spo_keypoints": [
    "Suspect_A -> observed at -> Crime_Scene_X",
    "DNA_Sample_B -> matched -> Defendant_C"
  ],
  "timestamp": "2026-10-14T11:49:18Z",
  "msp_id": "JudiciaryOrg1"
}

```

B. Paperless Similar Case Search and Integrity Checking

Jaccard keypoint similarity search. Legal researchers can search historical precedents by querying S-P-O keypoints against the CouchDB state index [12]. Smart contracts compute the Jaccard overlap coefficient between the query keypoint set and each stored record:

$$\mathfrak{Jaccard}(K_{\text{query}}, K_{\text{record}}) = \frac{|K_{\text{query}} \cap K_{\text{record}}|}{|K_{\text{query}} \cup K_{\text{record}}|} \quad (4)$$

On-demand integrity verification. If a party suspects that an evidence file or summary has been altered, the local document is re-hashed with SHA-256 [18] and the result is compared with the immutable digest stored on the ledger [11]. Changing even a single character changes the digest completely, so any alteration raises a tamper flag immediately.

VI. LEGAL COMPLIANCE AND ETHICAL CONSIDERATIONS

A. Data Protection Compliance (GDPR and DPDP Act)

Two data protection requirements matter most for this design [4, 16].

Right to erasure (GDPR Article 17). Raw case text and summaries live off-chain on IPFS, so an erasure order can be carried out by unpinning the off-chain payload or destroying its AES-256 encryption key [16]. What remains on-chain is a SHA-256 digest that no longer points to anything retrievable, which preserves ledger integrity. The EDPB points out, however, that a hash derived from personal data can itself remain personal data. It also says unsalted or unkeyed hashes are generally not considered sufficient protection, and that erasure works only if the off-chain data that would allow identification is deleted as well [16]. Salting data before hashing is also the practice described for Fabric's private data collections and purging off-chain data while keeping the on-chain hash may still fall short of full GDPR erasure [11]. A production deployment should therefore use salted or keyed hashes and obtain legal review for its own jurisdiction.

Automated redaction. LegNER masks personal identification details before files reach downstream models or are saved off-chain [4].

B. Human-in-the-Loop Judicial Oversight

A human stays in the loop, so that automated errors do not shape judicial outcomes [1, 7]. Temporal deltas, scene-affinity alerts and summaries support judges, prosecutors and investigators; they do not make decisions for them [1, 7].

VII. CONCLUSION AND FUTURE SCOPE

We have presented a framework for legal document processing, cold-case temporal analytics and cryptographic integrity verification. It pairs an off-chain hybrid NLP pipeline with on-chain Hyperledger Fabric smart-contract indexing and targets three practical problems: transformer hallucination, detecting evidence changes across multi-decade gaps, and paperless, tamper-evident precedent retrieval [1]. This paper sets out the design and its mathematical basis. It does not yet report results. A full empirical evaluation on benchmark corpora and real case collections must come before any performance claim can be made, and the system should be read as support for human judgment, not a replacement for it.

Future work starts with that evaluation. After that, we plan to explore zero-knowledge proofs (ZK-SNARKs), so that evidence can be verified across jurisdictions without revealing the underlying documents, and multi-modal transformers for video and audio evidence.

REFERENCES

- [1] A. Kanapala, S. Pal, and R. Gairola, "Summarization of legal documents: Where are we now and the way forward," *Computer Science Review*, vol. 40, p. 100388, 2021.
- [2] A. Shukla, P. Bhattacharya, S. Poddar, R. Mukherjee, K. Ghosh, P. Goyal, and S. Ghosh, "Legal case document summarization: Extractive and abstractive methods and their evaluation," in *Proc. 2nd Conf. Asia-Pacific Chapter Assoc. Comput. Linguist. (ACL)*, 2022, pp. 1048–1064.
- [3] W. Ansar et al., "LeSum: Cost-effective LLM-driven hybrid summarization of Indian legal judgments," *Expert Systems with Applications*, vol. 268, p. 126102, 2026.
- [4] I. Karamitsos, N. Roufas, K. Al-Hussaini, and A. Kanavos, "LegNER: A domain-adapted transformer for legal named entity recognition and text anonymization," *Front. Artif. Intell.*, vol. 8, Art. no. 1638971, 2025, doi: 10.3389/frai.2025.1638971.
- [5] J. Benet, "IPFS – Content addressed, versioned, P2P file system," arXiv:1407.3561, Jul. 2014.
- [6] E. Androulaki et al., "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Porto, Portugal, Apr. 2018, Art. no. 30, doi: 10.1145/3190508.3190538.
- [7] V. Oliveira, M. Queudot, et al., "AI-powered legal text summarization using LLMs for legal document analysis: Model performance evaluation," in *Proc. INCOFT, SCITEPRESS*, 2025, pp. 112–118.
- [8] J. Govea, I. Ortiz-Gárce, P. Palacios, A. Maldonado Navarro, S. Acurio Del Pino, and R. William, "Technical evaluation of language models adapted for the automation of legal contracts: Clause extraction, classification, and summarization," *Frontiers in Artificial Intelligence*, vol. 9, p. 1345892, 2026.
- [9] M. M. Rahman et al., "Natural language processing in legal document analysis software: A systematic review of current approaches, challenges, and opportunities," *Int. J. Innov. Res. Sci. Stud.*, vol. 8, no. 3, pp. 5026–5042, 2025.
- [10] S. Bonomi, M. Casini, and C. Ciccotelli, "B-CoC: A blockchain-based chain of custody for evidences management in digital forensics," in *Proc. Int. Conf. Blockchain Economics, Security and Protocols (Tokenomics 2019)*, ser. OASICS, vol. 71, 2020, pp. 12:1–12:15, doi: 10.4230/OASICS.Tokenomics.2019.12.
- [11] S. Natarajan, M. Sethi, D. Enyeart, Y. Manevich, and A. Barger, "Private data collections: Enabling data privacy in permissioned blockchains," *Proc. VLDB Endow.*, vol. 19, no. 12, pp. 4263–4275, 2026, doi: 10.14778/3827998.3828031.
- [12] Hyperledger Foundation, "CouchDB as the state database," *Hyperledger Fabric Documentation*, release 2.0. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.0/couchdb_as_state_database.html

- [13] L. Zhang, Y. Li, and H. Zhang, "Deep learning-based automatic summarization of Chinese maritime judgment documents," *Appl. Sci.*, vol. 15, no. 10, Art. no. 5434, 2025, doi: 10.3390/app15105434.
- [14] T. Sadikot and S. Jain, "Legal document summarization: A zero-shot modular agentic workflow approach," in *Proc. 1st Workshop NLP Empowering Justice (JUST-NLP)*, Mumbai, India, Dec. 2025.
- [15] S. U. Ahmed et al., "Smart contract generation through NLP and blockchain for legal documents," *Procedia Computer Science*, vol. 235, pp. 1420–1429, 2024.
- [16] European Data Protection Board, "Guidelines 02/2025 on processing of personal data through blockchain technologies," Version 2.0, adopted Jul. 7, 2026, Brussels, Belgium.
- [17] S. Saxena, "AI-based legal document summarization for judicial assistance," *Sci. J. Artif. Intell. Blockchain Technol.*, vol. 2, no. 3, pp. 71–80, Jul.–Sep. 2025, doi: 10.63345/sjaibt.v2.i3.309.
- [18] National Institute of Standards and Technology, "Secure Hash Standard (SHS)," Federal Information Processing Standards Publication FIPS PUB 180-4, Gaithersburg, MD, USA, Aug. 2015, doi: 10.6028/NIST.FIPS.180-4.
- [19] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," Federal Information Processing Standards Publication FIPS 197-upd1, Gaithersburg, MD, USA, May 2023, doi: 10.6028/NIST.FIPS.197-upd1.