PREPRINT SENT TO: ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE (2023) 000–000

Blockchain-Based Auditing of Legal Decisions Supported by Explainable AI and Generative AI Tools

Dr. Swati Sachan
Lecturer: Artificial Intelligence in Finance
University of Liverpool
Management School
Chatham St,
Liverpool, L69 7ZH
United Kingdom
Swati.Sachan@liverpool.ac.uk

Dr. Xi Liu (Lisa)
Lead Data Scientist
Kennedys Law LLP
16 John Dalton St,
Manchester, M2 6HY
United Kingdom
Lisa.Liu@kennedysiq.com

ARTICLE INFO

Keywords:
Legal
Law
Explainable AI
Blockchain
Generative AI
Responsible AI

ABSTRACT

Generative AI tools powered by Large Language Models (LLMs) have demonstrated advanced capabilities in understanding and articulating legal facts closer to the level of legal practitioners. However, scholars hold contrasting views on the reliability of the reasoning behind a decision derived from LLMs due to its black-box nature. Law firms are vigilant in recognizing the potential risks of violating confidentiality and inappropriate exposure of sensitive legal data through the prompt sent to Generative AI. This research attempts to find an equilibrium between responsible usage and control of human legal professionals over content produced by Generative AI through regular audits. It investigates the potential of Generative AI in drafting correspondence for pre-litigation decisions derived from an eXplainable AI (XAI) algorithm. This research presents an end-to-end process of designing the architecture and methodology for a blockchain-based auditing system. It detects unauthorized alterations of data repositories containing the decisions by an XAI model and automated textual explanation by Generative AI. The automated auditing by blockchain facilitates responsible usage of AI technologies and reduces discrepancies in tracing the accountability of adversarial decisions. It conceptualizes the two algorithms. First, strategic on-chain (within blockchain) and off-chain (outside blockchain) data storage in compliance with the data protection laws and critical requirements of stakeholders in a legal firm. Second, auditing by comparison of the unique signature as Merkle roots of files stored off-chain with their immutable blockchain counterpart. A case study on liability cases under tort law demonstrates the system implementation results.

1. Introduction

1.1 Background

The formulation and execution of laws entail information processing, logical reasoning, decision-making, and communication of legal decisions. Multiple levels of information processing make the legal sector an optimal domain for the application of Artificial Intelligence (AI) technologies (Prakken & Sartor, 2015). The substantial progress made in eXplainable AI (XAI) techniques and data accessibility has empowered the development of legal decision-support systems. These systems assist in legal decision-making without necessitating the replacement of lawyers with AI algorithms (Collenette, et al., 2023). A recent launch of Large Language Models (LLMs) such as Generative Pre-trained Transformer (GPT) by OpenAI (OpenAI, 2023) and Google Bard based on the Language Model for Dialogue Applications (LaMDA) by Google AI (Collins & Ghahramani, 2021) have stimulated significant interest in the legal

sector. The capacity of Generative AI technologies to produce human-like text has demonstrated the potential as indispensable tools for legal practitioners (Dwivedi, et al., 2023).

Law has emerged as a rich testing ground and a key area for Generative AI deployment. Scholars have tested ChatGPT's (version GPT-3.5) capabilities in passing actual law school examinations (Choi, et al., 2023). It exhibited a performance equivalent to a student averaging a C+ grade. Another study indicated that GPT-4 successfully cleared the Uniform Bar Exam, indicating that its automated legal reasoning is getting closer to human legal practitioners (Katz, et al., 2023). In light of these breakthroughs, legal educators are formulating strategies to confront the challenges brought by open-source Generative AI tools in legal education and professional practices (Ajevski, et al., 2023).

An investigative study on the legal drafting capabilities of ChatGPT has presented supportive results on its advanced capabilities in understanding simple facts and articulating the legal foundation of a case (Iu & Wong, 2023). The law upholds justice and societal stability. An algorithm cannot be held accountable for potential errors in high-stakes legal judgments. The legal community cannot proceed without resolving the complexities in the ownership and accountability of AI-generated content before it is considered seriously for legal drafting assistance and legal decision-making in the foreseeable future.

1.2 Data Security Concerns in Legal Practices

Legal firms are actively seeking an in-depth understanding of operational functionality and data usage strategy proposed for Generative AI tools developed by AI research laboratories such as Open AI and Google AI. Law firms are acutely aware of the potential risks of data breaches with the increased adoption of Generative AI technologies. The firms have raised three main concerns regarding its implementation: the inability to emulate legal reasoning, the potential fabrication of legal facts, and the mismanagement of confidential data and its misuse by malicious actors. In response to these concerns, law firms in the UK, USA, Canada, and several European countries have imposed restrictions on their lawyers' usage of tools, such as ChatGPT and Google Bard, as assistance in legal drafts or other queries (Reuters, 2023). Disclosing client-specific information, such as information of defendants, claimants, and other stakeholders, would violate the duty to maintain the confidentiality of the data subject's information. Legal professionals are explicitly advised against uploading client-specific case data onto AI platforms to prevent potential violations of confidentiality due to inappropriate exposure of sensitive data. This caution extended to other domains beyond the legal sector, such as finance (Bushard, 2023) and tech firms (Van Dis, et al., 2023).

The rise of Generative AI prominence has triggered the discussion on conflict of interest and ethics in the acquisition, storage, and utilization of user data, in addition to concerns surrounding data breaches due to unauthorized access to personally identifiable information. Establishing secured protocols for maintaining confidential information, including clear guidelines on its usage and association with users' accounts and identities, is crucial to engender trust in Generative AI tools (Dwivedi, et al., 2023).

Sensitive data is exposed through the prompt sent to Generative AI in two primary ways: user interaction through chatbots and APIs for direct access to a specific Generative AI algorithm. Platforms like ChatGPT and Google Bard provide options to delete chat activities; however, they automatically enroll users into their data collection system. According to the API data usage policy of OpenAI, the data transmitted through a code such as Python script is not used to train the LLMs. Despite these policies, data usage by Generative AI tools is still a relatively new and ambiguous area that requires further investigation and clarity.

1.3 Motivation on Implementation of a Blockchain-based Auditing System for AI

1.3.1 Repurpose Usability of Generative AI Tools

This research examines the practical usability of Generative AI to assist lawyers in brainstorming ideas while drafting legal correspondences. These correspondences communicate probable pre-litigation (out-of-court settlements) decisions derived from an XAI algorithm that supports lawyers in processing complex legal decisions**.** The rise of Generative AI raises two questions: Can these tools truly replace humans or specialized XAI algorithms trained for a specific application? Irrespective of their potential as replacements, how can we address the data security concerns, particularly data breaches due to sensitive data exposure through prompts?

The rationale behind the decisions by Generative AI powered by LLMs cannot be trusted entirely due to its black-box nature, and researchers hold contrasting views on its reasoning capabilities (Biever, 2023). At the same time, the probabilistic decisions by XAI, such as rule-based (expert systems) or data-driven models, are interpreted in terms of the importance of the features and activated rules. Lawyers are anticipated to comprehend the algorithmic output to make an informed legal decision, even when results are presented visually on front-end dashboards. It requires validation by the end users (Galanti, et al., 2023). Given these dynamics, this research repurposes the use of Generative AI tools in transforming the legal judgments derived by an XAI decision-support system into coherent textual explanations rather than direct handling of legal evidence and factual information. It anonymizes sensitive information in the decisions derived from the XAI model before sending it as a prompt to Generative AI. The generated textual explanations could assist lawyers in drafting legal correspondence within a law firm.

1.3.2 Integrity and Accountability for Responsible AI

The rapid advancements in AI require focused research efforts directed at ensuring its responsible application and enforcement of human authority over the deployment of these sophisticated technologies. Regular audits could confirm the control of human legal professionals over the usage of content by Generative AI tools and alignment of XAI decision-support systems with the established legal principles.

One potential way to accomplish this is by preserving the integrity of data repositories containing algorithmic decisions and textual explanations produced by Generative AI tools for future auditing by utilizing blockchain's immutability feature (permanent storage) to determine the accountability of incorrect legal decisions. For instance, misuse of AI tools could trace accountability back to human lawyers; lapses in algorithmic system maintenance and testing could point toward the negligence of data scientists or developers; and the occurrence of third-party malicious attacks indicates failure in data security.

1.4 Contribution

This research has presented an innovative end-to-end process for designing the architecture and conceptualizing the methodology for a blockchain-based auditing system. The system’s primary objective is to detect unauthorized alterations of data repositories containing the metadata of decisions by an XAI model and content produced by Generative AI tools. Figure 1 summarizes the motivation and contribution. The auditing practice helps to monitor the human usage of AI technologies and ensures data integrity to avoid discrepancies in tracing the accountability of adversarial legal decisions. It attempts to make the following contributions:

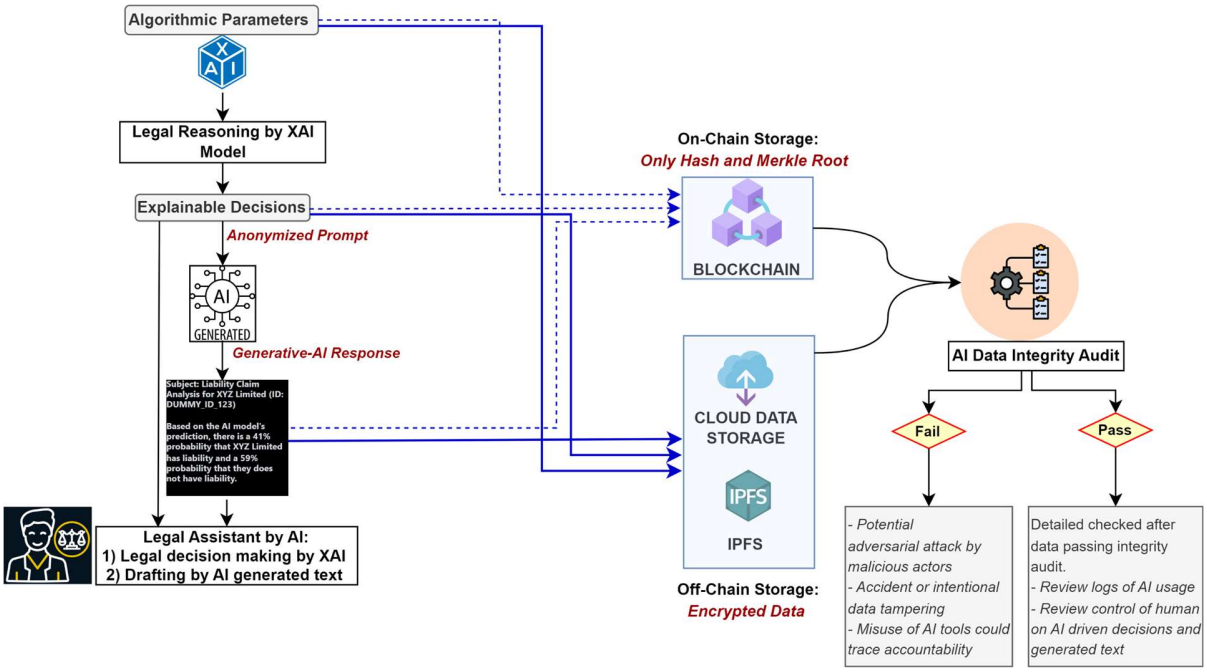

Fig. 1. Summarized Contribution and Motivation to Design Blockchain-Based Auditing System for AI

(a) Architectural Requirement of Blockchain-Based Auditing System: Requirement analysis is conducted to align stakeholders within a legal firm with the core design principles of the blockchain-centric auditing framework. It is a structured pathway for the identification of crucial objectives on compliance with data protection laws, the confidentiality of sensitive legal data, security through regular audits, and responsible usage of open-source Generative AI tools.

(b) Conceptualization of Blockchain-Based Auditing Process: The research conceptualizes the blockchain-based auditing process through two algorithms. The first algorithm presents a comprehensive strategy for on-chain (within blockchain) and off-chain (outside blockchain) storage of legal data in compliance with the data protection laws. The second algorithm outlines the auditing process, which compares the Merkle root (unique signature) of files stored in an off-chain location, such as Cloud storage, with its corresponding immutable Merkle root stored chronologically in a blockchain network.

(c) Explainable Legal Reasoning by Evidential Reasoning: An integrated explainable model based on Individual & Conjunctive Maximum-Likelihood Evidential Reasoning (I-MAKER & C-MAKER) is proposed to process ambiguous legal facts and heuristics to establish a causal relationship between evidence and the final hypothesis (decision).

(d) Case Study to Demonstrate System Implementation by Multiple Technology Integration: The proposed system's implementation results are demonstrated through a case study on vicarious liability arising from workplace accidents. The blockchain-based auditing framework is evaluated on two blockchain platforms: Ethereum (a public platform) and Hyperledger Fabric (a private platform). The optimal functionality and robust security of the auditing system are achieved through the integration of multiple technologies to mitigate each other shortcomings. The case study exemplifies the potential for responsible usage and trustworthy adoption of AI tools and emerging technologies in legal tasks.

1.5 Structure of the Paper

The rest of the paper is organized as follows. Section 2 presents the literature review on existing XAI frameworks on legal reasoning and proposed blockchain-based auditing techniques. Section 3 presents an end-to-end process of developing a blockchain-based auditing system to detect unauthorized alterations of data repositories containing metadata of decisions by an XAI model and automated textual explanation by Generative AI tools by malicious actors. Section 4 demonstrates the results of technological integration utilized to implement the proposed system in a case study based on employer liability cases arising from workplace accidents. The framework to audit the decisions derived from the XAI algorithm and their textual explanations produced by Generative AI tools is evaluated on two distinct blockchain platforms: Ethereum and Hyperledger Fabric. Section 5 addresses the limitations of the proposed system and its scope for future improvements. The paper is concluded in Section 6.

2. Literature Review

2.1 XAI Frameworks for Legal Knowledge Representation

AI researchers in the field of law have employed a logic-based structure for the representation of legal knowledge, particularly in legal argumentation. An in-depth survey illuminated the intersection of law and logic as a means to address legal reasoning ambiguities by natural language expression (Prakken & Sartor, 2015). Furthermore, it discussed the application of logic programming for the development of a legal knowledge-base. A computational argumentation model for explainable legal decision-support based on Abstract Dialectical Frameworks (ADF) is proposed for predicting judgment violation in the context of the European Court of Human Rights (Collenette, et al., 2023). ADF is a directed graph that represents and reasons with complex legal argumentation structures. This study compared the primary approach of legal reasoning based on HYPO (Rissland & Ashley, 1987), CATO (Aleven, 1997), and IBP (Bruninghaus & Ashley, 2003) with ADF (Al-Abdulkarim, et al., 2014). The HYPO, CATO, and IBP are pseudo-acronyms.

The Bayesian approach aligns with intuitive legal reasoning to link a single hypothesis and piece of evidence. However, real-world legal arguments have multiple hypotheses and evidence characterized by intricate causal dependencies (Fenton, et al., 2016). A study introduced a Bayesian model to address this issue (Neil, et al., 2019). It integrates independent legal arguments of two parties to express the guilt and innocence of the defendant. Further, a Bayesian network has been proposed for the articulation of legal syllogistic reasoning within the framework of statutory law interpretation (Constant, 2023).

An explainable deep-learning model was proposed for legal text summarisation by highlighting the relevant text based on attention score (Norkute, et al., 2021). However, it lacks comparative results on the interpretation of the highlighted text by model-agnostic methods such as (Local Interpretable Model-Agnostic Explanations) LIME (Ribeiro, et al., 2016) and Shapley additive explanations (SHAP) (Lundberg & Lee, 2017), as well as model-specific methods for deep learning such as Layer-Wise Relevance Propagation (LRP) (Binder, et al., 2016) and Deep Taylor decomposition (Montavon, et al., 2017). These interpretation methods are typically employed post hoc after the model has undergone training.

Despite AI advancements, a fully data-driven AI model, such as a deep neural network for legal reasoning, is not common due to the lack of explicit representation of legal facts and the need for explainability in high-stakes legal decisions. In legal claim handling, this gap was addressed by a hybrid

rule-based method, which assimilates human expert knowledge and data-driven training to offer transparent decisions for tort liability claims (Sachan, et al., 2021). It utilized the evidential reasoning (ER) approach to aggregate multiple belief rules activated by the circumstances of a legal case. ER is based on Dempster–Shafer (DS) theory of evidence, a general extension of Bayesian theory (Du & Zhong, 2021). This paper utilizes the I-MAKER & C-MAKER approach to process ambiguous legal evidence jointly (Sachan, et al., 2021).

2.2 Data Integrity Audit by Blockchain

Auditing data integrity through a centralized trusted Third-Party Auditor (TPA) poses reliability concerns due to single points of failure (Wu, 2016). Utilizing a TPA in dynamic environments, such as the Industrial Internet of Things (IIoT) and AI systems, which frequently interact with external technology service providers and various clients, is inadequate because of compromised data confidentiality and susceptibility to attacks (Liu, et al., 2017). The Provable Data Possession (PDP) mechanism proposed for the integrity verification of metadata poses issues with verification limits and high storage costs (Shah, et al., 2008). Another study introduced a similar mechanism for data verification and recovery (Juels & Kaliski Jr, 2007). However, PDP is vulnerable to attacks due to the centralized storage of user metadata. A study employed the Boneh–Lynn–Shacham signature mechanism to produce homomorphic verifiable signatures to reduce communication overhead and facilitate public auditing (Shacham & Waters, 2013). However, it falls short of guaranteeing user data privacy. Another integrity framework for Cloud storage failed to assure data confidentiality and security (Nepal, et al., 2011). A Cloud-based IoT data management scheme reduced the computational overhead of hash functions during the signature process but utilized random masking to maintain data privacy (Zhu, et al., 2019). These frameworks operate under the presumption of TPA's trustworthiness; however, in practice, the failure of a single TPA node can compromise the entire system (Wu, 2016).

The adoption of blockchain technology in industrial distributed ledger applications is driven by its immutability feature, and its implementation depends on three factors: scalability, performance, and maintenance (Siddiqui & Haroon, 2023) (Yang, et al., 2021). A blockchain-enabled method ensured data integrity verification by data owners and users, eliminating the dependence on TPA (Liu, et al., 2017). However, it lacks details on accomplishing data confidentiality. Another study introduced a hashing method to verify data integrity and emphasized the difficulties of maintaining data confidentiality on the blockchain (Zikratov, et al., 2017). But, it does not provide a strategy to maintain data confidentiality.

Another study introduced the decentralized collaborative verification system for multiple peers (Hao, et al., 2020). Each peer in the system keeps a complete verification record in the blockchain. Users can approach the collaborative network to retrieve verification outcomes by exploring their local blockchain. Despite these features, it persists in utilizing TPA for data integrity assessments and does not truly achieve decentralization. A study suggested the adoption of Merkle trees for data integrity verification for secured tamper-proof forest fire prediction communication from source to decision (Datta & Sinha, 2023). It does not address strategies for maintaining data confidentiality due to the non-sensitive nature of real-time forest fire prediction data. A study enhanced data confidentiality in a blockchain-based auditing system by leveraging Paillier homomorphic encryption for IIOT data (Zhang, et al., 2022). However, it does not address the on-chain and off-chain data storage strategies in compliance with the data protection laws and lacks details on cryptographic key management.

The framework presented in this research addresses both off-chain and on-chain data storage strategies for sensitive data and cryptographic keys. It robustly ensures data confidentiality by utilizing data encryption and hashing techniques.

2.3 Integration of Blockchain and AI

A literature review on accounting with blockchain technology and AI has recognized the impact of combining both technologies to develop a coherent ecosystem for advanced auditing systems. The auditing system can leverage blockchain's immutable and verifiable features and AI's ability to learn from data to apply augmented decision-making (Han, et al., 2023). A comprehensive bibliometric and literature analysis review concerning the application of blockchain as a security layer for AI-based systems recognized the need for more in-depth research on the effective implementation of successful and stable integration of blockchain within AI systems (Shinde, et al., 2021).

A theoretical framework proposed a complex AI system for decentralized consensuses of multiple XAI predictors managed by a smart contract to reach a final decision (Nassar, et al., 2020). A study proposed auditing XAI decisions by storing in IPFS due to storage limitations on Ethereum. However, it failed to demonstrate the robustness test with a use case and the experimental results on blockchain performance metrics such as throughput and latency (Malhotra, et al., 2021). A study introduced an integrated approach combining blockchain technology with an Explainable Deep Neural Network (x-DNN) for use in medical indemnity insurance. This system employed blockchain to securely document the consent for sharing medico-legal data among various entities, such as insurance companies, law firms, and hospitals. The aggregated data from the blockchain then feeds into the x-DNN for "lawyer-in-the-loop" decisions. Lawyers interpreted the decision by analyzing the visual interpretation of x-DNN decisions by LRP, LIME, and SHAP (Sachan & Muwanga, 2023). Another research employed blockchain to integrate the knowledge of multiple experts to formulate lending criteria for small business loans. This approach aimed at reliable lending decisions to ensure financial inclusivity for underserved communities (Sachan, et al., 2023).

Centralized AI systems are susceptible to cyber-attacks. Minor modifications to machine learning algorithm parameters can potentially compromise the AI system's performance. A research study employed the intrinsic immutability of blockchain technology to create a secure storage and auditing system specifically for the parameters of convolutional neural networks designed to identify defects in manufacturing processes (Song & Moon, 2021). A paper proposed a similar principle for future research direction on integrating Blockchain and AI for robust protection of Autonomous Vehicles against malicious attacks (Bendiab, et al., 2023). Furthermore, another study employed blockchain's capabilities to ensure resilience against malicious activities and handle conflicting traffic incident event reports (Philip & Saravanaguru, 2023). A dual-stage Long Short-Term Memory (LSTM) for event prediction and Bayesian for event conflict resolution model was proposed to resolve conflicts arising from trusted event sources. The system thoroughly cross-references and authenticates the event before transmitting it to a blockchain as evidence.

Table 1 presents the definitions of the key blockchain concepts used in this paper for clear understanding across diverse communities, such as legal professionals, AI researchers, and blockchain enthusiasts.

Table 1: Key Blockchain Concepts for Blockchain-Based Auditing

Concept	Definition
On-Chain Data	The data is stored and verified directly in a blockchain network as a transactional record.
Off-Chain Data	The data is not stored directly in a blockchain network. Instead, it is stored in external systems or databases.
Hashing	Hashing is a computational process that transforms data into a unique, fixed-size string of characters known as a hash. The slightest modification to the data changes the hash value, indicating a potential tampering attempt by a malicious actor (Ünsal, et al., 2023).
Encryption	Encryption is the process of converting sensitive data into a ciphertext form using an encryption algorithm and secret keys. It is used in blockchain to secure sensitive data by making it unreadable to unauthorized parties (Das, et al., 2021) (Joe & Raj, 2021) (Agyekum, et al., 2021).
Smart Contract	It is a self-executable contract with predefined rules and conditions written as a code inside blockchain (Taherdoost, 2023) (Kushwaha, et al., 2022).
IPFS (Interplanetary File System)	IPFS is a decentralized file system that stores and shares off-chain data, like metadata and images, across a peer-to-peer network (Muralidharan & Ko, 2019). It achieves decentralization by loading content from thousands of peers instead of relying on a single centralized server. This distributed approach eliminates single points of failure and control. Each piece of data in IPFS is cryptographically hashed into a secure content identifier (CID) to ensure data integrity, uniqueness, and reliable content retrieval and verification.
Difference between IPFS and Blockchain	IPFS is a decentralized file system and content-addressable network protocol designed to replace Hypertext Transfer Protocol (HTTP) for efficient file storage and sharing. It focuses on content-based addressing. In contrast, blockchain is a distributed ledger technology used for maintaining a decentralized record of transactions with consensus mechanisms and smart contract functionality (Nizamuddin, et al., 2018) (Kang, et al., 2022). Both contribute to decentralization but have different architectures and purposes.
Blockchain Nodes	Every blockchain is composed of multiple nodes. A node is a computer with an IP address that creates, sends, and receives blockchain data. Users engage with the network through these nodes, which serve as communication endpoints. Following are the three main node types: (a) **Full Nodes**: A blockchain node typically refers to a full node by default. It connects to the blockchain server of a decentralized network (Ray, et al., 2020). It validates new blocks and maintains a blockchain's transaction history, stores, copies, and distributes data (transaction) across the network. (b) **Pruned Full Nodes**: Pruned nodes support a full node and prioritize security over storage (Huang, et al., 2022). It first downloads the entire blockchain to its hard drive, then gradually deletes older data blocks, starting from the earliest. (c) **Archival Full Nodes**: An archival full node hosts the complete blockchain database (Ray, et al., 2020). It preserves the entire history of the blockchain network to ensure that all data is readily available to users for queries.

3. Methodology

This section presents an end-to-end process of designing architecture and conceptualizing the automated blockchain-based auditing system. It leverages the immutability feature of blockchain, which detects tampering in legal decisions by XAI and monitors content produced by Generative AI tools to promote responsible AI practices.

3.1 Architectural Requirement of Blockchain-Based Auditing System for Legal Practices

Figure 2 presents the architecture of a blockchain-centric framework to audit the data files containing decisions derived from XAI models and automated content produced by Generative AI tools. It is designed to safeguard the data from potential tampering and monitor the application of AI-generated content to assist human lawyers in drafting legal correspondences. These correspondences can be emails, letters, or system updates within a law firm to outline the expected decision of a legal case against a defendant. The requirement analysis for the successful integration of multiple technologies with blockchain platforms from the stakeholders' perspective in a legal firm is shown in Table 2. It demonstrates the importance of the architecture's compliance with data protection laws applicable to blockchain deployments and AI models. It points out the objectives such as confidentiality of sensitive legal data, quality assurance and security through regular audits, and reliable usage of open-source Generative AI tools.

Table 2: Requirement Analysis for Blockchain-Based Auditing System

Key Requirements	Design Aims
R1: Compliance with Data Protection Laws	Compliance with Article 5(1)(c) of the General Data Protection Regulation (GDPR), known as the "Data Minimization Principle," ensures the retention of only the necessary sensitive data (GDPR Article 5, 2023).
	Compliance with the GDPR's Article 17, "Right to be Forgotten," ensures the right to the permanent removal of personal data and prohibits data retention without explicit consent (Union, 2016) (Finck, M., 2019) Compliance with the California Privacy Rights Act 2020 (CPRA), particularly regarding the "Right to Rectification," aligns with the principles of GDPR (Sunyaev, 2020) (Karisma & Moslemzadeh Tehrani, 2023).
	Compliance with GDPR's Article 22, "Right to explanation," ensures transparency in algorithmic decisions by providing meaningful logic behind each decision (Union, 2016).
R2: Confidentiality and Integrity	Secure storage of the primary stakeholder's identity in the law firm
	Secure storage and control mechanisms for legal data access
R3:Quality Assurance	Regular audits or spot checks to verify the authority of human lawyers in the usage of automated content created by Generative AI tools
	Verification of the usage of AI-based systems in alignment with the legal principles and precedents
R4: Robustness and Security	Robust model architecture to withstand adversarial attacks
	Regular monitoring of the XAI system
	Maintain detailed records of algorithmic parameters and architecture.
R5: Regulatory Audits	Support for the collaboration between internal and external audits.
R6: Scalability and Efficiency	Scalable architecture designed to meet current and future demands by optimizing throughput and latency for increased transaction volumes.

The proposed architecture has three core components: a front-end web application, a back-end API server encompassing multiple modules, and a hybrid on-chain (within blockchain) and off-chain (outside blockchain) data storage mechanism.

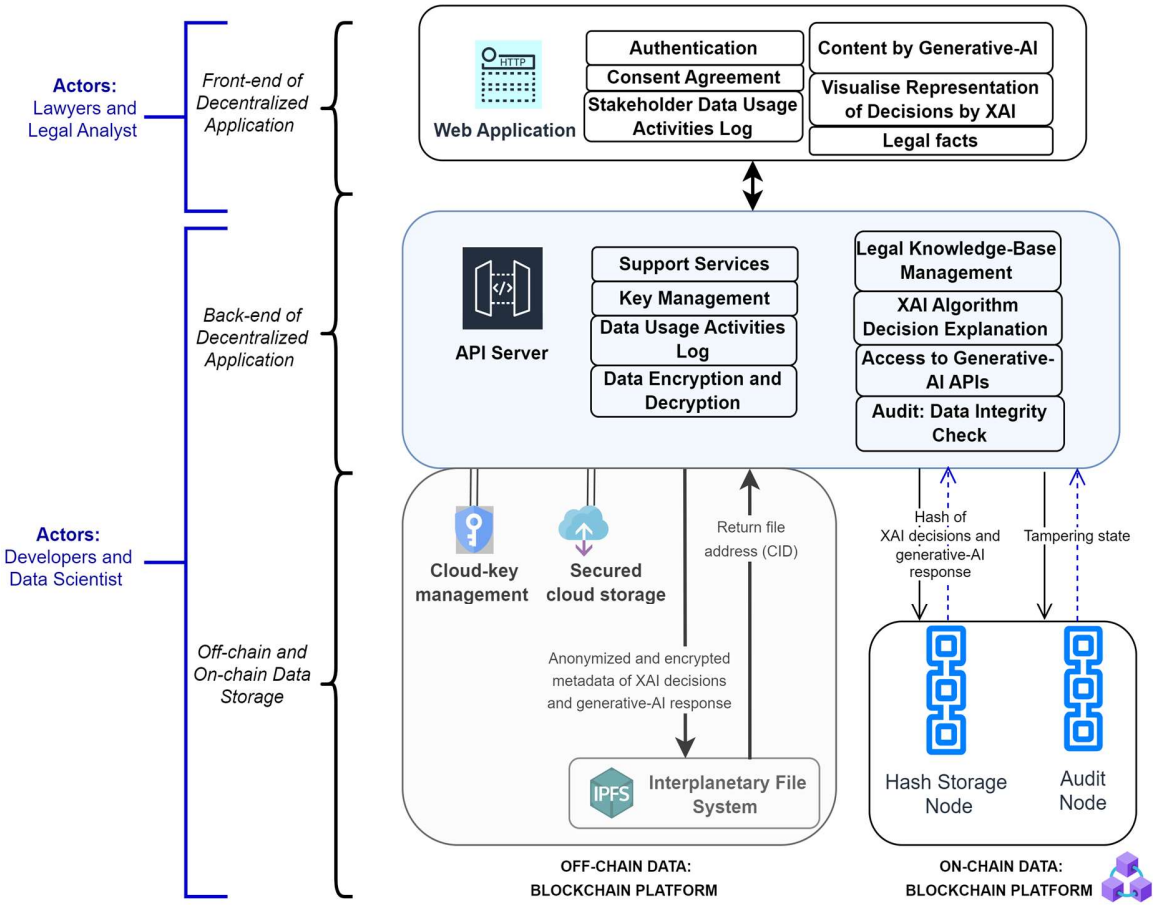

Fig. 2. Architecture for blockchain-based auditing system

3.1.1 Front-end Application: A front-end application is an interactive dashboard for users such as lawyers and legal analysts within a law firm. It functions as an information management tool that communicates with the API server to handle user authentication, data consent and monitor data usage activities (Sachan, et al., 2023). It presents the content produced by Generative AI tools, a visual analysis of decisions made by an XAI model, and a set of legal facts of a case. This research primarily concentrates on the management and regulatory compliance related to the storage of on-chain & off-chain data and access control mechanisms of XAI and Generative AI.

3.1.2 Back-end API Server: The API server ensures interoperability between multiple services: a front-end application, a blockchain platform, cryptographic-key storage service providers, secure Cloud data storage service providers, and an IPFS service providers to store encrypted and anonymized metadata of XAI algorithm and content produced by the Generative AI tools.

The Proxy Re-Encryption (PRE) scheme is employed for the encryption and decryption of data by legitimate users without revealing any information to intermediaries (Manzoor, et al., 2021) (Hasan, et al., 2020). A data file of a legal case after the completion of a decision-making task aided by an XAI model and content from a Generative AI tool is denoted by $D_{x,h}$ where x represents a legal case and h represents a human lawyer. In the PRE scheme, a human lawyer acts as a delegator, and an internal or external auditor serves as a delegate, denoted by a.

The pair of public and secret keys for both the delegator and delegate is represented as (pk_h, sk_h) and (pk_a, sk_a), respectively. The data file $(D_{x,h})$ is encrypted by the delegator's public key, yielding $\overline{D_{x,h}} = Enc(pk_h, D_{x,h})$. This encrypted data is subsequently uploaded to an off-chain database. An auditor as delegate requests decryption permission from the delegator by notifying his public key (pk_a). A re-encryption key $(rk_{h\to o})$ is generated specifically for a delegate if a delegator approves the access of legal files of a case. It is sent to a proxy module to re-encrypt the previously encrypted data; $\overline{\overline{D_{x,h}}} = ReEnc(rk_{h\to o}, \overline{D_{x,h}})$. The delegate (auditor) then receives this re-encrypted data and uses their secret key to decrypt it to obtain the original data; $D_{x,h} = DeCr(sk_a, \overline{\overline{D_{x,h}}})$.

Cryptography key management is a challenge in blockchain applications. The responsibility of key management falls on the users themselves. Blockchain networks rely on third-party storage providers for cryptographic key protection (Wöhrer & Zdun, 2021) (Hasan, et al., 2020). The API server is connected to the third-party cryptographic storage service provider and hosts a support services module for user registration and management. For instance, we used Azure Key Vault to test the proposed system. The back-end API server has an additional module to manage data access logs. It hosts AI-based modules for legal facts and statutes management, centralized access control to Generative AI APIs, and XAI algorithms within a legal firm.

3.1.3 Off-Chain and On-Chain Data Storage: This architecture integrates a hybrid off-chain and on-chain data storage strategy to ensure compliance with data protection laws to overcome the problem of permanent storage of sensitive data in an immutable ledger.

In this framework, IPFS serves as a repository for encrypted and anonymized metadata of XAI models and content generated by Generative AI. It is a decentralized and distributed file storage system designed to replace HTTP for more efficient file storage and sharing. Each data chunk in the IPFS system is cryptographically hashed, generating a secure Content Identifier (CID) for data integrity and uniqueness. Generally, IPFS files are paired with blockchain implementations for off-chain storage of actual files, while only hash pointers to those files are kept on the blockchain because it can hold large-sized files compared to the blockchain.

The data in both IPFS and Blockchain is immutable. The efficient protocols for delegated content erasure from IPFS files are proposed to align it with the critical data protection laws (Politou, et al., 2020) (Politou, et al., 2022). However, the decision to upload only encrypted and anonymized data in the IPFS is to support the risk appetite of legal firms.

In this framework, encrypted keys, user IDs, and legal case IDs are stored in an external Cloud key storage service, whereas secure Cloud data storage services handle the law firm's internal data. The copy of data in the Cloud is stored in IPFS; only hash pointers of Cloud and IPFS files are stored on the blockchain for audit purposes, as shown in Table 3. The blockchain environment is designed with two independent full nodes: the hash storage node and the audit node.

(a) Hash Storage Node: The hash storage node is responsible for storing the hash values of decisions by XAI, algorithmic metadata, and content produced by Generative AI tools for a legal case.

(b) Audit Node: The audit node maintains records of the tampered state of the files as the outcome of the regular automated auditing by blockchain.

Each node maintains its own copy of data on the blockchain network to ensure audit integrity and promote the scalability of the blockchain application. The partitioning of data in multiple nodes manages computational and storage workloads.

Table 3: Off-chain and On-chain Storage

Storage Location	Storage Medium	Data Type	Functionality
Off-chain	**Cloud**	Encrypted Data	Cloud data is accessed frequently by internal stakeholders such as developers and domain experts (legal professionals). However, it is vulnerable to unintentional tampering and malicious attacks.
	Cloud Key Management	Cryptographic Keys, user IDs, and legal case IDs	Utilizes external Cloud service providers to safeguard cryptographic keys and essential IDs (User and case credentials).
	IPFS	Encrypted and Anonymized Data	IPFS files store the XAI algorithmic decisions & parameters and Generative AI responses. IPFS files are paired with the Cloud and Blockchain as a scalable solution for storing large off-chain data and recovery of original untampered files if Cloud storage is compromised. It enables robust auditing for data integrity checks and accountability in AI decisions.
On-chain	**Blockchain**	Hash values of files	Only the Hash pointers of files in IPFS and Cloud are stored in the blockchain for data integrity audits for protection against misuse of AI.

A detailed process of auditing the tampering of data from Generative AI tools and XAI algorithms will be discussed in Section 3.4. Before this, the methodology for formulating legal decisions using XAI and secure usage of Generative AI tools is presented in Sections 3.2 and 3.3.

3.2 Legal Decisions by MAKER Approach

3.2.1 Introduction to Legal Knowledge Representation by MAKER

The implementation of logic within legal contexts is driven by the need for an explicit and accurate representation of legal norms compared to its conveyance in commonly used natural language by legal practitioners such as lawyers, legislators, and jurists. Logic acts as a tool for resolving inherent ambiguities in the natural language expression of legal arguments by representing it into a set of logical axioms for unambiguous relations (Prakken & Sartor, 2015). For instance, the logical-linguistic structure of legal facts in an employer liability case: "An employer is liable for a workplace accident if they failed to provide protective equipment and/or a witness testifies to the accident," can be represented by clear and unambiguous logic as:

$$(Protective\ Equipment(x) = No) \vee (Witness\ Testimony(x) = Yes) \rightarrow Liability(x) \quad (1.1)$$

$$(Protective\ Equipment(x) = No) \wedge (Witness\ Testimony(x) = Yes) \rightarrow Liability(x) \quad (1.2)$$

In Expressions (1.1) & (1.2), the logic connective "$\vee$" and "$\wedge$" represents the "AND" and "OR" operations, respectively. These logical syntaxes are also called legal syllogisms, where categorical syllogisms of premises are used to infer a conclusion (Constant, 2023). The logical syntax shown in Expression (1.1) represents an organization (defendant) denoted by x, liability towards its employee

(claimant) for a workplace accident if they failed to provide protective equipment "AND" a witness testifies to the accident. Alternatively, Expression (1.2) points to the ultimate hypothesis of an organization's liability, if they failed to provide protective equipment "OR" a witness testifies to the accident.

The Evidential Reasoning (ER) (Yang & Xu, 2013) (Fu, et al., 2015) based framework is employed to capture legal facts and heuristics to facilitate the establishment of a causal relationship between evidence and the final hypothesis or conclusion. ER is based on Dempster–Shafer's theory of evidence (Dempster, 2008). The Individual Maximum-Likelihood Evidential Reasoning (I-MAKER) methodology is utilized to quantify the extent to which a specific piece of legal evidence supports the state of a legal fact by considering the strength of the evidence. A legal fact is represented by q such that $q \in \{1, \dots, Q\}$ and each legal fact can have V_q number of categorical values to represent the legal circumstances such that $v \in \{1, \dots, V_q\}$. Therefore, the piece of evidence is denoted by $e_{v,q}$. A decision, in legal terms a final verdict or ultimate hypothesis, is represented by θ.

The strength of the evidence is assessed by two crucial parameters: the weight and reliability of the evidence toward a conclusion. The weight of evidence refers to the importance of evidence for a given decision, denoted by $w_{\theta,v,q}$. The reliability of the evidence refers to the credibility of the source of evidence for a given decision, denoted by $r_{\theta,v,q}$. The integration of the weight and reliability concept of ER adopted in the MAKER framework provides a comprehensive explanation of decisions (Liu, et al., 2019). The I-MAKER and C-MAKER extend the MAKER framework to pre-process ambiguous evidence (Sachan, et al., 2021).

The I-MAKER is utilized to process an individual piece of ambiguous legal evidence, and the C-MAKER combines multiple pieces of evidence to draw a decision by estimating the joint probability mass. For instance, C-MAKER can estimate the joint probability mass of evidence for $Protective\ Equipment = No$ and $Witness\ Testimony = Yes$ in Expression (1.1) for a final liability decision. Table 4 provides the preliminary definitions of terms used for legal reasoning in the context of the MAKER model.

Table 4: Preliminary Definitions for Legal Reasoning by MAKER

Legal facts (q)	It is the data or information used by lawyers as a foundation for their arguments for legal precedents and statutes. The establishment of facts is based on the evaluation of evidence.
Conclusion or decision (θ)	It is the set of decisions or ultimate verdicts of a legal case.
Evidence ($e_{v,q}$)	Evidence encompasses the data or information used to validate the facts supporting a legal argument. It helps to determine whether the facts are sufficient, insufficient, or inconclusive for the case at hand.
Weight of evidence ($w_{\theta,v,q}$)	It is the significance or importance of a piece of evidence in shaping a conclusion or final decision.
Reliability of legal evidence ($r_{\theta,v,q}$)	It is the credibility or trustworthiness of the source from which a piece of legal evidence originates.
Probability mass to evidence ($m_{\theta,v,q}$)	It is the extent of support for a legal fact by a piece of evidence toward a conclusion.

3.2.2 Processing of Legal Evidence by I-MAKER

A dataset has N number of legal cases with $q, q \in \{1, \dots, Q\}$ attributes for legal fact representation to evaluate the legal liability. Each attribute has $v \in \{1, \dots, V_q\}$ referential values to represent the legal fact circumstances. The target attribute (θ) is a set of possible decisions defined as the frame of discernment $\Theta = \{\theta_1, \dots, \theta_z, \dots, \theta_Z,\ z \in \{1, \dots, Z\}\}$. These decisions are mutually exclusive and collectively exhaustive. The decision by MAKER is provided over the power set of the frame of discernment:

$$P(\Theta) = \{\emptyset, \{\theta_1\}, \dots, \{\theta_Z\}, \dots, \{\theta_1, \dots, \theta_{Z-1}\}, \Theta\} \tag{2}$$

Uncertainty of a piece of evidence is quantified by the number of samples supporting a decision. The degree of support for a decision is called the belief-degree. The belief-degree for a piece of evidence is obtained by estimating the probability mass for a v^{th} circumstance of a q^{th} legal fact for a decision $\theta \in P(\Theta)$. Evidence $\left(e_{v,q}\right)$ is profiled over a belief distribution and the sum of belief is equal to one, as follows:

$$e_{v,q} = \left\{\left(e_{\theta,v,q}, \hat{m}_{\theta,v,q}\right), \forall \theta \in P(\Theta)\right\} \tag{3.1}$$

$$\sum_{\theta \in P(\Theta)} \hat{m}_{\theta,v,q} = 1 \tag{3.2}$$

The probability mass normalized by weight and reliability of evidence for a given legal claim is:

$$\hat{m}_{\theta,v,q} = \begin{cases} 0 & \theta = \emptyset \\ \frac{m_{\theta,v,q}}{(1+w_{\theta,v,q}-r_{\theta,v,q})} & \theta \subseteq \Theta, \theta \neq \emptyset \\ \frac{(1-r_{\theta,v,q})}{(1+w_{\theta,v,q}-r_{\theta,v,q})} & \theta = P(\Theta) \end{cases} \tag{4}$$

Here, $m_{\theta,v,q}$ is the basic probability mass and $\hat{m}_{\theta,v,q}$ is the normalized probability mass of evidence $e_{v,q}$ for a decision θ. The basic probability mass is calculated by the normalization of the likelihood of evidence (Sachan, et al., 2021). The weight and reliability of evidence could be a subjective judgment of lawyers. It can be trained by data-driven optimization. The objective function to optimize the weight and reliability of each piece of evidence is:

$$\begin{gathered} Minimize{:}f(w_{\theta,v,q},r_{\theta,v,q}) = \frac{1}{2N}\sum_{i=1}^{N}\sum_{\theta \in P(\Theta)}\left(m^o - \hat{m}(w_{\theta,v,q},r_{\theta,v,q})\right)^2 \\ constraints{:}0 \leq w_{\theta,v,q} \leq 1, 0 \leq r_{\theta,v,q} \leq 1 \end{gathered} \tag{5}$$

The observed probability for an instance x is denoted by m^o and $\hat{m}(w_{\theta,v,q}, r_{\theta,v,q})$ is the estimated normalized probability mass.

3.2.3 Aggregation for Legal Evidence by C-MAKER

A single piece of evidence processed by I-MAKER is aggregated by the C-MAKER approach to infer a final decision (Sachan, et al., 2020) (Liu, et al., 2019). The normalized joint probability mass of joint pieces of evidence from two mutually exclusive attributes of legal facts q and q' is denoted by $\hat{m}_{\theta,v,v'}$,where $v \in V_q$ and $v' \in V_{q'}$ denote the v^{th} and v'^{th}, such that $q \neq q'$ and $q, q' \in \{1, \dots, Q\}$. The interrelation index between two evidences $e_{v,q}$ and $e_{v',q'}$ pointing to class h and h', with $h \cap h' = \theta, \forall \theta \in P(\Theta)$, respectively, is given as follows:

$$\psi_{\theta,v,v'} = \begin{cases} 0 & ;if\ \ m_{h,v,q} = 0\ or\ m_{h',v',q'} = 0 \\ \dfrac{m_{\theta,v,v'}}{m_{h,v,q}\ \ m_{h',v',q'}} & otherwise \end{cases} \quad (6)$$

In the above Expression (6), two pieces of evidence are deemed independent if $\psi_{\theta,v,v'} = 1$ and disjoint if $\psi_{\theta,v,v'} = 0$. The joint probability mass ($\widehat{m}_{\theta,v,v'}$) for a decision θ is supported by evidence $e_{v,q}$ and $e_{v',q'}$ is given by:

$$\widehat{m}_{\theta,v,v'} = \begin{cases} 0 & \theta = \emptyset \\ \dfrac{m_{\theta,v,v'}}{\sum_{\theta \in P(\Theta)}\left[m_{\theta,v,v'} + \left(m_{\Theta,v,q}\ m_{\Theta,v',q'}\right)\right]} & \forall \theta \in P(\Theta), \theta \neq \emptyset \end{cases} \quad (7a)$$

$$m_{\theta,v,v'} = [(1 - r_{\theta,v',q'})\ m_{\theta,v,q} + (1 - r_{\theta,v,q})\ m_{\theta,v',q'}] + \sum_{h \cap h' = \theta} \gamma_{h,h',v,v'}\ \psi_{h,h',v,v'}\ m_{h,v,q}\ m_{h',v',q'} \quad (7b)$$

The normalized joint probability mass, presented in Equation (7a), is derived from Equations (7b). and (7c). The Expression $\left(m_{\Theta,v,q}\ m_{\Theta,v',q'}\right)$ in Equation (7a) is the residual support of the power set. The parameter $\gamma_{h,h',v,v'}$ known as the reliability ratio, is the proportion of the joint reliability of the two pieces of evidence to the product of their individual reliabilities (Sachan, et al., 2020).

The decision generated by the MAKER algorithm is articulated in a structured, human-readable text format, encapsulated within a JSON (JavaScript Object Notation) file, an example shown in Figure 3. An anonymized text version of a legal case's explanation is merged with a fixed prompt to send in a Generative AI tool to get a response.

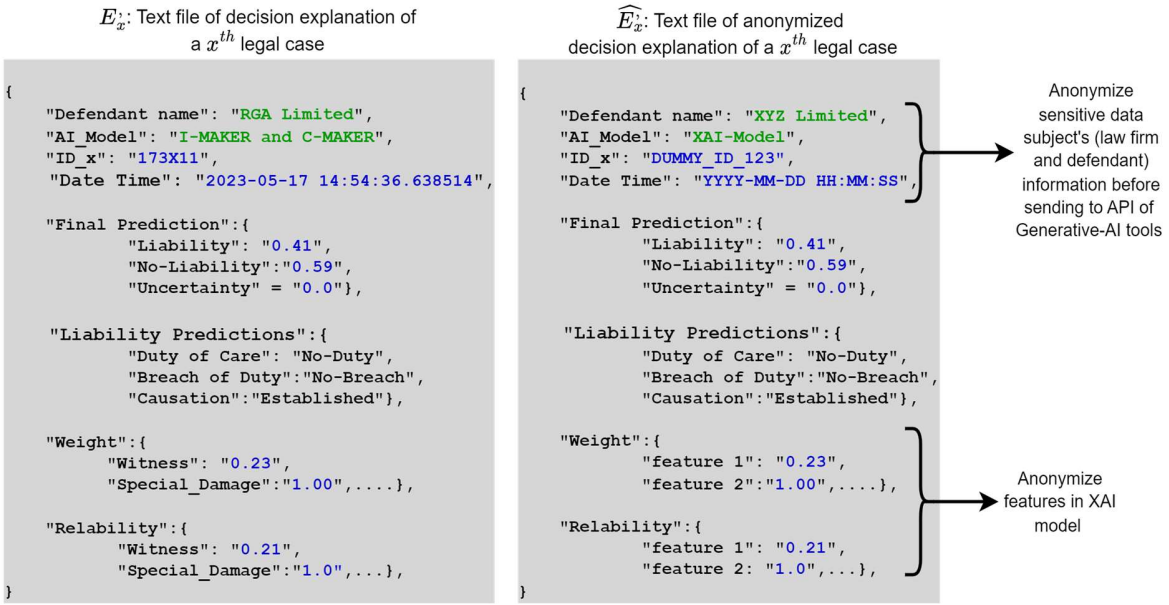

Fig. 3. Example of decision generated by I-MAKER & C-MAKER model in a text format

3.3 Drafting XAI Decisions by Generative AI

The XAI model generates an explanation denoted as E_x for a legal case against an individual or defendant x. This explanation is subsequently formatted into a text file, denoted by E_x'. To preserve

privacy, an anonymized version of the file is created by removing the data subject's identifiable information and substituting it with placeholder or dummy data. The data subject could be the defendant and the respective law firm. The anonymized text file denoted by $\hat{E}_x'$ is concatenated with a standardized prompt, denoted by P_x. An example of an anonymized text file is shown in Figure 3. The standardization implies the removal of any identifiable or internal information related to the law firm in the prompt statement, such as queries originating from internal email communications. The content generated as a response to the prompt through the OpenAI API and Google Bard-API is denoted as R_x. Figure 4 illustrates the dialogue between the prompt and the Generative AI API's response.

A sequence of prompts is dispatched to the API of four prominent large-language models: Bard, gpt-text-davinci-003, gpt-text-davinci-002, and GPT-4 to generate content. This action is executed by a programming language code (for example, a Python script), which utilizes unique API keys to obtain responses. The ethical premises and assumptions for the application of Generative AI tools in legal firms are:

(a) Confidentiality of Prompt Data: The limitations on prompt data are enforced to maintain confidentiality by anonymizing sensitive information. Ethical data handling practices and privacy is maintained by anonymizing the output of the XAI model used for content generation, which excludes personal data, precise case specifics, and third-party information. This approach safeguards the transmission of sensitive information in the Generative AI platforms.

(b) Restriction on Lawyer's Direct Access: Computing devices managed by law firms restricts the direct access of advanced chatbots such as ChatGPT and Google Bard. Instead, the content is generated by API initiated at a lawyer's request to ensure the anonymity of prompt information. This restriction circumvents the potential misuse of open-source AI tools.

3.4 Blockchain for Auditing XAI Decisions and AI-Created Content

The concept of an 'immutable ledger' represents the unalterable and tamper-resistant feature of blockchain technology. The record of a cryptographic hash of the information into the blockchain creates a unique and irreversible footprint of the actual data stored outside the blockchain. The hash of algorithmic decisions by XAI and content produced by large-language models supported by Generative AI tools is stored in the blockchain. The data stored outside the blockchain on a firm's centralized server is susceptible to tampering by malicious actors.

The immutability of a blockchain ledger serves as a trustworthy repository to provide documented evidence of sequential activities. It provides a reliable referencing of the prompt and its corresponding text produced by Generative AI tools, thereby allowing for meticulous examination of AI-produced content employed by a lawyer. The verification process can pinpoint which portion of the generated text was incorporated or excluded by the lawyer during the drafting of correspondence. This scrutiny can help answer questions like, *"To what extent did the lawyer use the AI-generated text?"*.

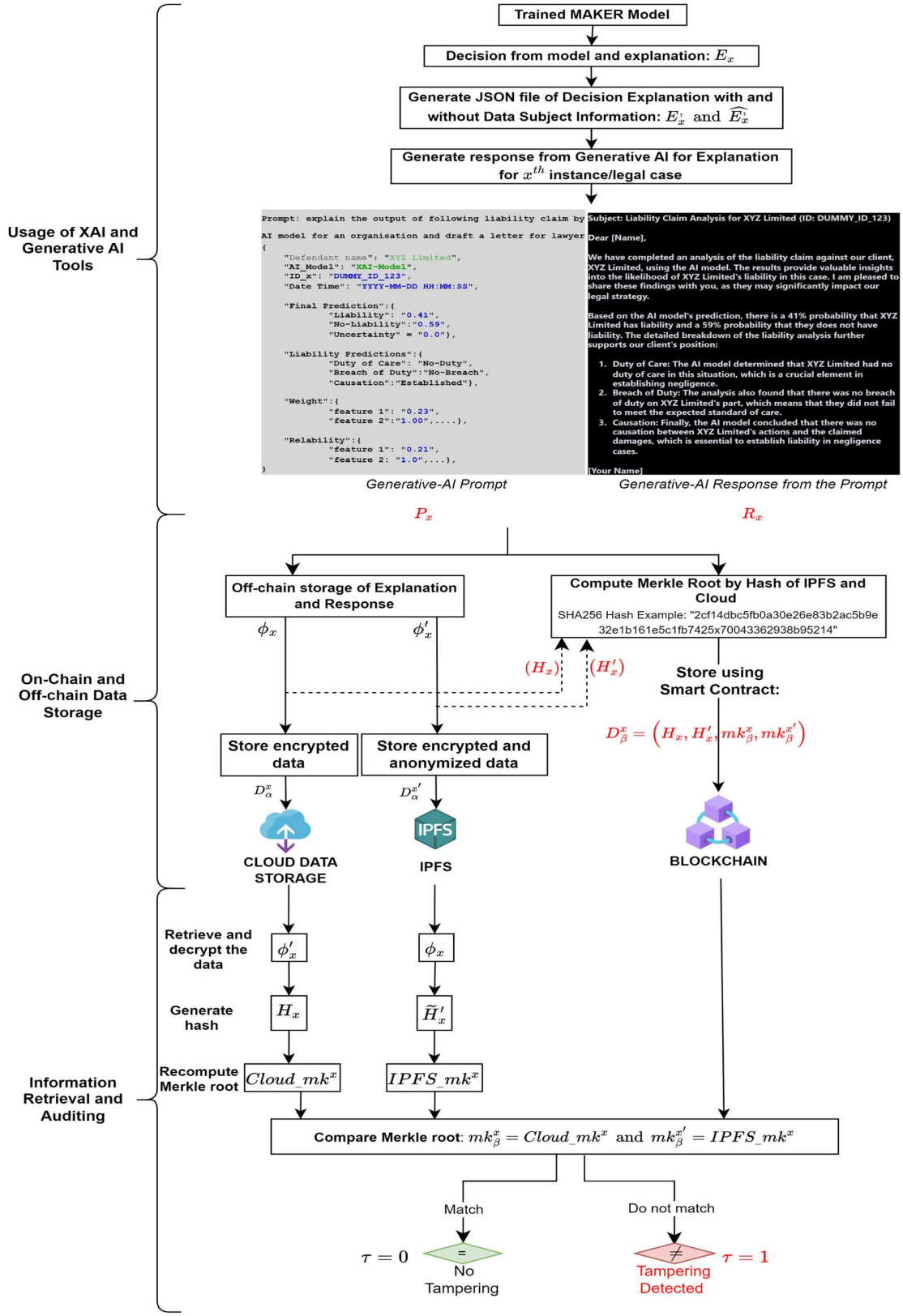

Fig. 4. Blockchain-based auditing process for legal drafting and response management

Additionally, blockchain technology can improve accountability and trust in legal decisions made by XAI models by maintaining algorithmic data integrity. Figure 4 illustrates the three processes. First, obtaining legal decisions by the XAI model and its usage in producing a textual explainable of decisions by Generative AI tools, such as GPT and Bard. Second, off-chain and on-chain storage mechanisms. Third, blockchain-based the auditing procedure.

The on-chain & off-chain data storage and auditing processes are subdivided into algorithm 1 (Table 5) and algorithm 2 (Table 6), respectively. Algorithm 1 records data in the hash node, and algorithm 2 in the audit node of a blockchain network. Both nodes are registered in a blockchain network. A successful transaction to record data in a node requires node authentication by a smart contract (or chain code). The authentication ID is denoted by $Stored\_Auth\_ID$. It can be generated by creating the hash of node ID and digital signature:

$$Stored\_Auth\_ID = SHA256(Node_{ID}, D_{sign}) \quad (8)$$

Here, $Node_{ID}$ is the hash value of the node ID and D_{sign} is the digital signature of a valid user. A smart contract validates a node by comparing its authentication ID with a regenerated hash value. The hash value is calculated by the SHA256 hashing algorithm (Martino & Cilardo, 2020).

3.4.1 Algorithm 1

Algorithm 1 in Table 5, presents the steps to store anonymized and encrypted legal decisions from XAI and its corresponding response by Generative AI tools to an off-chain (α) and on-chain (β) data storage platforms to facilitate the process of automated auditing by blockchain. The chosen off-chain platforms are Cloud and IPFS. The on-chain data is stored in blockchain networks such as Ethereum and Hyperledger.

Step 1: Anonymize XAI Decision File

The MAKER model generates an explanation (E_x) for a decision (θ) for a legal case against a defendant (x). This explanation is then saved as a JSON file (E'_x). The choice of the XAI model may differ among law firms; in this research, MAKER is utilized to provide explainable legal decisions. The JSON file is anonymized by substituting the identifiable information with a piece of dummy information $(\hat{E}'_x)$ to ensure compliance with the data protection law against permanent storage in decentralized platforms such as IPFS and leakage of sensitive information to Generative AI tools.

Step 2: Send Anonymized Prompt to Generative AI

The plain text of the anonymized JSON file is concatenated with the standardized prompt statement (P_x) to prepare a query for Generative AI. The prompt is sent to the API of Generative AI rather than copy-paste to their chatbots for better control and customization to obtain a response (R_x).

Step 3: Encrypt and Anonymize Data for Off-chain Storage

The original data files are stored in an encrypted format within IPFS files and the Cloud for centralized control. Only the respective hash values of these files are retained in the blockchain. For Cloud storage, a data file is prepared by combining the hash of a legal claim ID (H_{ID}), the non-anonymized JSON file of explanation, and the response obtained from the Generative AI query as $\{H_{ID}, E'_x, R_x\}$ into a single JSON file (ϕ_x). In parallel, the data intended for IPFS is compiled by merging the hash of a legal claim ID, the

anonymized JSON explanation file, and the response to the Generative AI's query as $\{H_{ID}, \hat{E}'_x, R_x\}$, into another unique JSON file (ϕ'_x). Then, the hash value of the files: ϕ_x and ϕ'_x is generated, denoted as H_x and H'_x, respectively.

Step 4: Compute Merkle root and Hash to Prepare On-chain Data Storage

Merkle tree is the authentication tree that creates the unique signature of the independent data stored chronologically in the blockchain (Datta & Sinha, 2023) (Hariharasitaraman & Balakannan, 2019). It validates the integrity of data stored within an individual block and across a series of blocks within the chain. The Merkle Tree is a binary tree where each node can have zero, one, or two child nodes. It first forms the hash of the leaf nodes in pairs, and the next tree level is created by hashing the concatenated hashes of two child nodes. This process continues until a single node remains at the top of the tree called Merkle root.

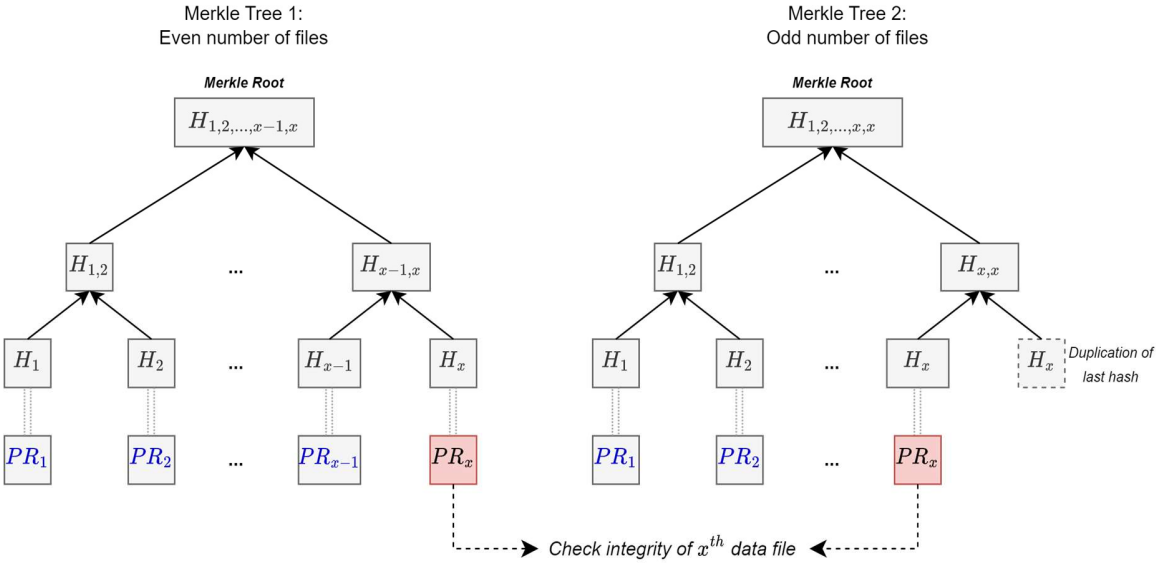

Fig. 5. Merkle Tree with even and odd numbers of files stored in the hash storage node

A Merkle root for both Cloud and IPFS is calculated for each x^{th} legal file associated with a defendant. This computation utilizes the hash values of previous IPFS and Cloud files stored sequentially in the blockchain, where the previous file number in the sequence can be represented as $\{1, \ldots, x-1\}$. A Merkle root is a combination of hash values of off-chain data stored on the blockchain up to legal case x. An illustration of a Merkle tree with an odd and even number of files is presented in Figure 5. The hash value is duplicated if a Merkle tree has an odd number of nodes at any level. The Merkle root derived by computing the Merkle tree of the hashed Cloud file and hashed IPFS files is denoted as mk_x^{β} and $mk_{x'}^{\beta}$, respectively. These Merkle roots for Cloud and IPFS are distinct due to the differing nature of data storage. The data held on IPFS is anonymized, unlike that on Cloud, resulting in distinct hash values, which provide different Merkle roots.

Step 5: Encode Off-chain and On-chain data

The hash value of data stored in the Cloud (H_x) and hash value of data stored in the IPFS file (H'_x), Merkle root of Cloud $\left(mk_x^{\beta}\right)$ and IPFS $\left(mk_{x'}^{\beta}\right)$ are consolidated into a single JSON file. This file is then converted into the BASE64 format to prepare the data for on-chain storage $\left(D_{\beta}^{x}\right)$. The data D_{β}^{x} is pushed inside a block of the hash storage node of a blockchain network, and a successful transaction (i.e. data upload) returns the transaction ID, represented as $TH_{hash\_node}^{x}$.

The conversion of data to BASE64 text format is a common practice, as it is compatible with the 256-bit slot available in both IPFS and Blockchain. The BASE64 can encode binary streams such as images, videos, and text for reliable transmission of binary information. Encryption and decryption are used to hide something (a secret message) while encoding and decoding are used to bring a piece of information into a specific form.

Step 6: Upload Prepared Data to Off-chain and On-chain Storage

Once the fingerprint of off-chain data as hash values is securely stored in the blockchain, the corresponding files of a legal case, both pre-anonymized and post-anonymized, are saved respectively in the Cloud and IPFS. As an additional security measure, the files residing in the Cloud and IPFS are first encoded into the BASE64 text format and then encrypted using the data owner's public keys under the PRE scheme. The resulting off-chain encrypted files, D_x^{α} and $D_{x'}^{\alpha}$ are uploaded to the Cloud and IPFS, respectively. A successful upload to IPFS will return a file identifier, denoted as CID_x.

3.4.2 Algorithm 2

Algorithm 2 in Table 6, presents the procedure to audit the integrity of data files by ensuring the consistency of off-chain data files in the Cloud and IPFS. The key principle of the algorithm is to compare the locally recomputed Merkle roots of the files in off-chain storage facilities (IPFS and Cloud) with their corresponding Merkle root stored on the blockchain.

Step 1: Initialization

The algorithm runs for a series of X legal cases, recorded chronologically on the blockchain, where each case is represented by a specific index $x \in \{1, \dots, X\}$. The audit process begins with the first legal file ($x = 1$) in the block after the genesis block and proceeds sequentially up to the last block containing the most recent legal file ($x = X$).

The process begins with the initialization of three empty lists: '$Tampered_{files}$' for any files found tampered, '$cloud_{hash}$' and '$IPFS_{hash}$' for recalculated hash values of the files in the Cloud and IPFS, respectively.

Step 2: Retrieve and Decrypt Off-chain Encrypted Data

For every legal case file $x \in \{1, \dots, X\}$ the algorithm first retrieves the encrypted and encoded files stored in secure Cloud storage (D_x^{α}) and the IPFS $\left(D_{x'}^{\alpha}\right)$. With the data owner's (such as the case processing lawyer) permission, these files are decrypted and decoded to extract the Cloud and IPFS files (ϕ_x and ϕ'_x).

Step 3: Hash and Merkle Root Recomputation

The SHA256 hash value of these decrypted files is recalculated to generate hashes for the Cloud data files $(\widetilde{H_x})$and the IPFS files $(\widetilde{H'_x})$. These hashes are then appended to their respective lists: '$cloud\_hash$' for Cloud files and '$IPFS\_hash$' for IPFS files. The recalculated hash values of the files retrieved from the Cloud and IPFS are added to their respective lists: $cloud\_hash \leftarrow [\widetilde{H_1}, \dots, \widetilde{H_{x-1}}, \widetilde{H_x}]$ *and IPFS files:* $IPFS\_hash \leftarrow [\widetilde{H'_1}, \dots, \widetilde{H'_{x-1}}, \widetilde{H'_x}]$. The hash values in these lists are utilized to recompute the Markle root for the files stored in the Cloud and IPFS denoted as $Cloud\_mk^x$ and $IPFS\_mk^{x'}$, respectively.

Step 4: Validate Data Integrity by Merkle Root Comparison

The previously stored Merkle Roots for the Cloud data $\left(mk_x^\beta\right)$ and IPFS files $\left(mk_{x'}^\beta\right)$ are retrieved from the full hash node of the blockchain. The data of an x^{th} legal case in the full hash node is located by the transaction ID $\left(TH_{hash\_node}^x\right)$ returned by Algorithm 1. Also, entire data transmitted by a full node can be downloaded from the computer acting as the server.

Any discrepancies between the newly computed Merkle root and its blockchain counterpart, indicate potential data tampering with the off-chain data, indicating a breach of data integrity. Whereas, strict equality between Merkle roots, affirms data integrity. The tampering state can be represented as:

$$\tau = \begin{cases} 1, \ mk_\beta^x = Cloud\_mk^x \text{ and } mk_{x'}^\beta = IPFS\_mk^{x'} \\ 0, \ other\ wise \end{cases} \tag{9}$$

In Expression (8), τ represents the tampering state. This condition provides a binary flag for data integrity, with '1' signifying data is intact and '0' signifying potential tampering. After all the files have been checked, the algorithm proceeds to document the legal data's integrity status in the audit node. This creates a permanent record of an audit outcome for a given legal case. The audit node within the blockchain maintains an immutable and traceable historical record of audit activities with timestamps to provide a means to retrospectively verify the integrity of data at any given point in the past.

The audit process for the metadata of the algorithmic parameters follows an identical procedure. The blockchain-based verification allows continuous monitoring of the algorithm. Regular auditing by cross-referencing the metadata of an XAI model, such as parameters, enables the detection and prevention of adversarial attacks.

3.4.3 Computational Complexity

(a) Algorithm 1: The execution time of the tasks in an algorithm is defined by Big O notation. The legal decision is derived from pre-trained I-MAKER and C-MAKER by pre-processing each attribute in the data and estimating joint probability mass, respectively. The computation complexity of I-MAKER is $O(Q)$ and C-MAKER is $O(Q!)$ (Sachan, et al., 2021). Here, Q is the number of attributes in a dataset and $Q!$ represents the joint pieces of evidence across multiple attributes. Data anonymization, concatenating anonymized data with a prompt statement, API call to generate text, a hash of fixed-size input data, encryption-decryption, and off-chain storage processes are scaled linearly, operate in constant time $O(1)$. The time complexity in computer science refers to how the runtime of an algorithm grows relative to its input size. The time required to store data in the blockchain depends on network latency and throughput. Detailed experiments on Ethereum and Hyperledger networks are documented in Section 4. The time complexity to compute the Merkle Root from a Merkle Tree is $O\big(Log(n)\big)$, where n is the number of nodes (hashes) in the tree. The significant time complexity of this algorithm is the execution of C-MAKER, $O(Q!)$.

(b) Algorithm 2: The initialization step has a constant time complexity $O(1)$. This loop runs X times where X is the number of legal case files in the blockchain network. Retrieval of encrypted JSON data from Cloud storage and IPFS, decryption of each file, and computation of SHA256 for each file have a constant time complexity $O(1)$ because each JSON formatted file has approximately the same size. Similarly, a comparison of two Merkle Root has a constant time $O(1)$.

For auditing, the Merkle tree for Cloud and IPFS file data is reconstructed and computed locally, then compared with the immutable Merkle root stored in the blockchain. The Merkle Tree time complexity is $O(Log(n))$, where n is the number of nodes (hashes) in the tree. The total time complexity to compute Merkle Root for an auditing iterative loop increases with the increases in the number of hashed files as $O(X * Log(X))$. Therefore, the most significant time complexity of algorithm 2 is $O(X * Log(X))$.

Table 5: Algorithm 1

Algorithm 1: Store XAI Decisions and Responses by Generative AI to IPFS file (off-chain), secured Cloud (off-chain), and a blockchain (on-chain)

Input:
Decision θ for a legal case x by an XAI algorithm
Prompt statement represented as $Prompt\_Statment$
Legal case ID as x_{ID}
Hash value of data files stored in the blockchain before legal case x: $\{H_1, \dots, H_{x-1}\}$
Authentication ID provided by a user trying to access the network: $Provided\_Auth\_ID$

Output:
On-chain data in Hash Storage Node: ID of successful transaction in blockchain ($TH^{x}_{hash\_node}$)
Off-chain data: Content identifier of the IPFS file (CID_x)

1.	*// Generate an explanation E_x of the decision of a legal case x by XAI model* $E_x = \text{XAI}\ (x, \theta)$
2.	*// Convert the explanation E_x into a JSON file for structured storage* $E'_x = JSON(E_x)$
3.	*// Remove sensitive information from the JSON file for privacy* $\hat{E}'_x = JSON(E'_x)$
4.	*// Create a comprehensive prompt by concatenating the anonymized JSON file to the given prompt statement* $P_x \leftarrow concatenate(Prompt\_Statment, \hat{E}'_x)$
5.	*// Generate a response from Generative AI by passing the P_x to API* $R_x \leftarrow Generative\_AI\_API(P_x)$
6.	*// Calculate a SHA256 hash of the legal case's unique identifier* $H_{ID} \leftarrow SHA256(x_{ID})$
7.	*//Cloud (ϕ_x) and IPFS file (ϕ'_x) preparation: Merge the information into a single JSON file* $\phi_x \leftarrow merge\_into\_JSON\{H_{ID}, E'_x, R_x\}$ $\phi'_x \leftarrow merge\_into\_JSON\{H_{ID}, \hat{E}'_x, R_x\}$
8.	*// Compute SHA256 hash of merged JSON files for Cloud and IPFS* $H_x \leftarrow SHA256(\phi_x)$ $H'_x \leftarrow SHA256(\phi'_x)$
9.	*// Compute the Merkle Root using hash values of previous IPFS and Cloud files stored chronologically in the blockchain along with the hash of current legal case files* $mk^{\beta}_{x} = compute\_Merkle\_Root(H_1, \dots, H_{x-1}, H_x)$ $mk^{\beta}_{x'} = compute\_Merkle\_Root(H'_1, \dots, H'_{x'-1}, H'_{x'})$
10.	*// Prepare data for blockchain storage* $D^{x}_{\beta} \leftarrow BASE64\left(merge\_into\_JSON\left\{H_x, H'_x, mk^{x}_{\beta}, mk^{\beta}_{x'}\right\}\right)$
11.	**IF** $Stored\_Auth\_ID == Provided\_Auth\_ID$ **THEN:**
12.	*// Store the BASE64 encoded data to the blockchain and retrieve the ID of the successful transaction*

	$TH^{x}_{hash\_node} \leftarrow \boldsymbol{Store\_to\_Blockchain\_Hash\_Storage\_Node}(D^{X}_{\beta})$
13.	**ELSE:**
14.	Invalid Transaction and Potential Attack by Malicious Actor
15.	**END**
16.	*// Encode Cloud (ϕ_x) and IPFS (ϕ'_x) data in BASE64 format for universal data transfer* $\alpha_x \leftarrow BASE64(\phi_x)$ $\alpha_{x'} \leftarrow BASE64(\phi'_x)$
17.	*// Encrypt the BASE64 encoded JSON file using the owner's cryptographic keys for additional security* $D^{\alpha}_{x} \leftarrow Encrypt(\alpha_x)$ $D^{\alpha}_{x'} \leftarrow Encrypt(\alpha_{x'})$
18.	*// Upload encrypted files to secure Cloud storage and IPFS, and retrieve the unique content identifier (CID) of the IPFS file* $x_{ID} \leftarrow \boldsymbol{Store\_to\_Cloud}(D^{\alpha}_{x})$ $CID_x \leftarrow \boldsymbol{Store\_to\_IPFS}(D^{\alpha}_{x'})$
19.	*// Return the transaction ID and CID of the IPFS file* **RETURN** $TH^{x}_{hash\_node}, CID_x$
20.	**END**

Table 6: Algorithm 2

Algorithm 2: Blockchain-based Audit Trail for Data Integrity

Suppose we have X legal case files stored as hash values in the blockchain network, where x ∈ {1, … , X}. An auditor's task is to monitor the integrity of these files using a blockchain-based audit trail.

Input:
Content identifier of IPFS file: CID_x
Legal case ID stored in the Cloud: x_{ID}
Authentication ID provided by a user trying to access the network: $Provided\_Auth\_ID$

Output:
List of Tampering files: $Tampered\_files$
Blockchain transaction ID for successfully recording the tampered state of files: $TH^{x}_{audit\_node}$

1.	*// Initialization: Create empty lists for the tampered files and recalculated hash of files in the Cloud and IPFS* $Tampered\_files \leftarrow []$ $cloud\_hash \leftarrow []$ $IPFS\_hash \leftarrow []$
2.	**For** $x = 1$ *to* X **do**:
3.	*// Retrieve data from Cloud storage and IPFS using respective IDs* $D^{\alpha}_{x} \leftarrow retrieve\_Cloud\_file(x_{ID})$ $D^{\alpha}_{x'} \leftarrow retrieve\_IPFS\_file(CID_x)$
4.	*// Decrypt the Cloud and IPFS files with the data owner's permission* $\alpha_x \leftarrow Decrypt\big(Decode(D^{\alpha}_{x})\big)$ $\alpha_{x'} \leftarrow Decrypt\left(Decode\big(D^{\alpha}_{x'}\big)\right)$
5.	*// Extract data files from decrypted and decoded Cloud and IPFS files* $\phi_x \leftarrow retrive(\alpha_x)$ $\phi'_x \leftarrow retrive(\alpha_{x'})$
6.	*// Recalculate the hash of retrieved files to recompute the Merkle root* $\widetilde{H_x} \leftarrow SHA256(\phi_x)$ $\widetilde{H'_x} \leftarrow SHA256(\phi'_x)$
7.	*// Append recalculated SHA256 hash values of files retrieved from Cloud and IPFS to respective lists* $cloud\_hash \leftarrow append\big(\widetilde{H_x}\big)$ $IPFS\_hash \leftarrow append\big(\widetilde{H'_x}\big)$

8.	*// Recompute the Merkle Root using hashes of all Cloud and IPFS files up to the x^{th} file:* $Cloud\_hash \leftarrow [\widetilde{H_1}, \dots, \widetilde{H_{x-1}}, \widetilde{H_x}]$ *and IPFS files:* $IPFS\_hash \leftarrow [\widetilde{H'_1}, \dots, \widetilde{H'_{x-1}}, \widetilde{H'_x}]$ $Cloud\_mk^x \leftarrow recompute\_Merkle\_Root(cloud\_hash)$ $IPFS\_mk^{x'} \leftarrow recompute\_Merkle\_Root(IPFS\_hash)$
9.	*// Get the Merkle Roots of IPFS files and Cloud data of x^{th} instance stored in the blockchain* $mk^x_\beta, mk^\beta_{x'} \leftarrow get\_Blockchain\_Hash\_Storage\_Node(TH^x_{audit\_node})$
10.	*// Validate the computed Merkle root against the Merkle root retrieved from the blockchain* **IF** $mk^x_\beta = Cloud\_mk^x$ and $mk^\beta_{x'} = IPFS\_mk^{x'}$ **THEN:**
11.	*// If the Merkle roots match, flag the case as not tampered* $\tau = 1$
12.	**ELSE:**
13.	*// If the Merkle roots do not match, flag the case as a tampering event* $\tau = 0$
14.	*// Store identification of tampered legal files: hash value of legal file ID and CID of IPFS file* $Tampered\_files \leftarrow append([H_{ID}, H_x^{CID'}])$
15.	**END**
16.	**END**
17.	**IF** $Stored\_Auth\_ID == Provided\_Auth\_ID$ **THEN:**
18.	*// Store the latest tampering state of the legal data to audit the node of the blockchain* $TH^x_{audit\_node} \leftarrow Store\_to\_Blockchain\_Audit\_Node(Tampered\_files)$
19.	**ELSE:**
20.	Invalid Transaction and Potential Attack by Malicious Actor
21.	**RETURN** $Tampered\_files, TH^x_{audit\_node}$
22.	**END**

3.5 Security and Integrity Evaluation of Blockchain-Based Audit Framework

Blockchain stores the hash pointers of the files stored in Cloud and IPFS, which contain encrypted and anonymized metadata of AI outcomes. Blockchain platforms inherently possess commendable security features. However, no technology is entirely immune to cyber threats. The following lemma presents proof of the robustness of the proposed Blockchain-based auditing framework:

(a) Challenge 1: How credible are blockchain-audit outcomes, especially when malicious actors intend to tamper with data stored within the blockchain, such as the hash pointers of Cloud and IPFS files?

Lemma 1: The proposed framework resists tampering with on-chain data against any unauthorized modifications

Proof of High Tampering Complexity: The architecture of a blockchain is characterized by a series of blocks interconnected by the cryptographic hash value of the preceding block. This creates an immutable link, where each block is connected to all its previous blocks in the chain. Each network participant needs an authentication ID, which is the joint hash of the Node ID and digital signature of a valid user, as shown in Expression (8), to confirm a successful transaction (or data record) within a block. An external secured Cloud Key Service Provider manages these IDs and cryptographic keys.

If malicious actors attempt to tamper with the data within a block, they would need to modify all subsequent blocks linked in the blockchain. Even if they succeed in changing all the blocks, the digital

signature will become invalid, and peers in the network will receive a notification on the data irregularity. The cost and complexity of tampering with block data increase proportionally to the blockchain's length.

Unlike traditional centralized systems, a blockchain operates under a decentralized model. It is distributed across peer-to-peer networks of nodes, which are continuously synchronized and updated. It lacks a single point of failure and is immune to changes made from a single computer. Massive computational power is required to successfully access and manipulate a chain of data in the blockchain network.

For instance, there are b_x number of blocks after the x^{th} block in a blockchain, and there are j nodes managing this blockchain. The computational power required by malicious entities to tamper the T_x number of blocks stored in the distributed network of nodes is:

$$T_x = j + j^2 + \cdots + j^{b_x} = \frac{j^{b_x+1} - j}{j-1} \tag{10}$$

For example, if there are 8 blocks after the 5^{th} block ($b_5 = 8$) in a network of 21 nodes ($j = 21$), the malicious entities would have to alter approximately $3.97e^{10}$ blocks to successfully tamper with the data.

(b) Challenge 2: Can the system withstand attempts by malicious actors who can synthetically engineer hash collisions with authentic blockchain hash values to bypass the audit process?

Lemma 2: The utilized hashing function has a strong defense against collision scenarios.

Proof of Negligent Hash Collision Risks: The proposed framework conducts the data integrity audit by comparing the Merkle root of the files retrieved from the off-chain storage (IPFS and Cloud) with the Merkle root stored on the blockchain. The Merkle root is derived from the combination of the hash values of the preceding files stored chronologically in the blockchain. The security of the hash function largely depends on the ability to resist collisions. A collision happens when two distinct data, D and D', $D \neq D'$, generate an identical hash output, such that $hash(D) = hash(D')$.

Off-chain data is susceptible to unauthorized alterations. However, the proposed audit mechanism can detect tampering with off-chain data, which compares the locally computed Merkle root with the Merkle root retained in the blockchain. Audit failure will occur if the hash value of two distinct off-chain collides with on-chain data and vice versa. Therefore, the security of the hash function must be thoroughly evaluated to check whether an attacker could locate a pair of hash collisions due to the weakness of the hash function.

Birthday Attack Evaluation: The birthday attack is named after the birthday paradox. It can be used to approximate the probability of encountering collisions in a hash function during random attack attempts (Conrad, et al., 2016). Assume a hash function that produces an m-bits message digest, and the maximum possible number of unique hashes that it can generate equals is $n = 2^m$. If X random values are generated, then the probability of getting at least one pair of a hash collision by birthday paradox is:

$$P(X, n) = 1 - e^{\frac{-X(X-1)}{2n}} \tag{11}$$

For instance, if a 22-bit hash value is generated 300 trillion times, the probability of experiencing a hash collision is approximately one in 100 billion (Zhang, et al., 2022). However, this framework uses the SHA-256 hash function, which produces a 64-bit hash value. The potential for hash collisions with a 64-bit hash is significantly lower than with a 22-bit hash. Given the enormous space of its possibilities, the likelihood of a collision is virtually negligible.

4. Pilot Test Implementation on Liability Cases

4.1 Explainable Legal Decisions by I-MAKER and C-MAKER Approach

A tort is defined as a civil wrong that inflicts harm upon a claimant (Seavey, 1942). Under tort law, employers can be held liable for negligence of duty toward the physical and psychological well-being of the employees during their course of employment (Torrey & McIntyre, 2015). This case study demonstrates the application of the proposed methodology on employer liability cases arising from workplace accidents.

The legislation has formulated the general abstract term for legal norms to understand the circumstances of a legal claim (Prakken & Sartor, 2015). Three fundamental elements ascertain an organization's liability towards its employees. First, "duty of care" requires employers to maintain a safe working environment (Davies, 1989). Second, a "breach of duty" occurs when an employer's actions or inactions deviate from the expected standard of care (Sykes, 1988) (Bell, 2013). Third, "causation" is the proof of negligence (Morris, 1952). It is established when the employee experiences injury or financial loss as a direct result of the employer's breach of duty. These three fundamental elements are applicable across a broad spectrum of tort categories. Therefore, the proposed technique can be adapted to economic, property, dignitary, strict & absolute liability, and nuisance torts claims.

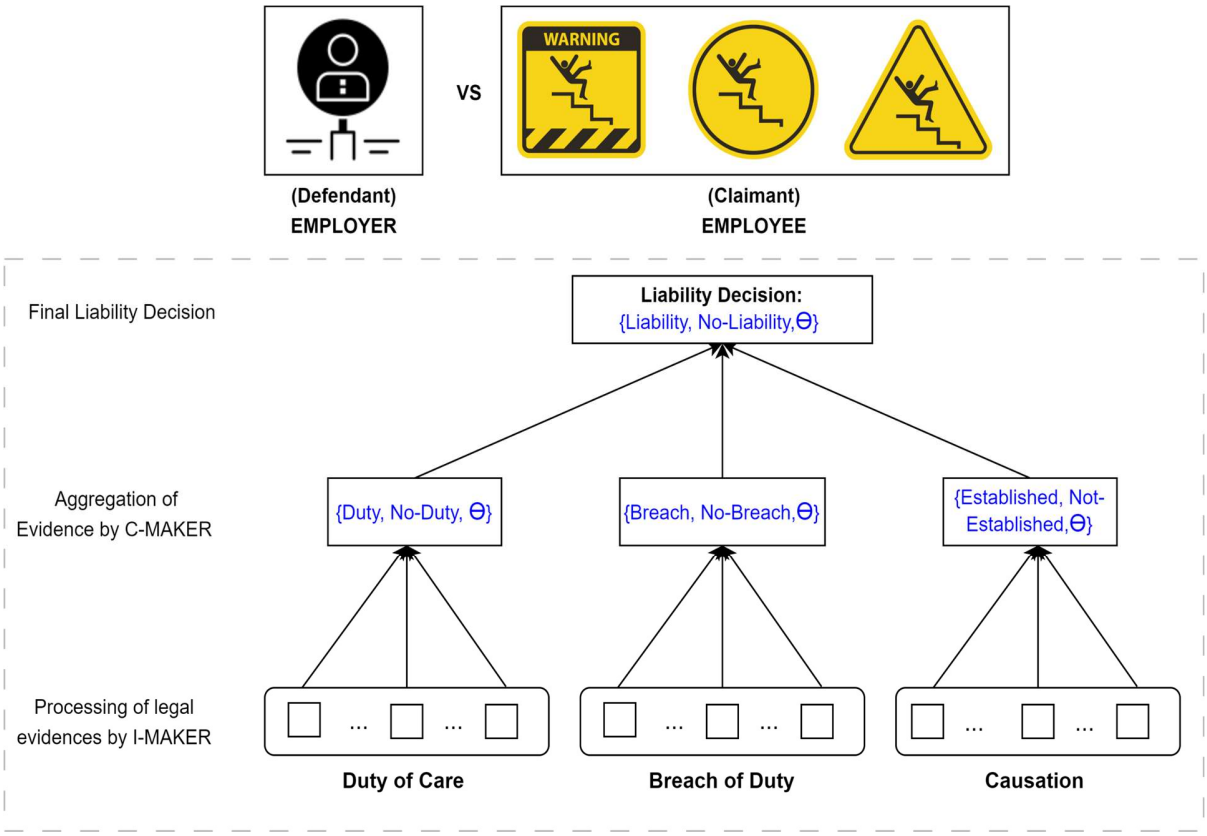

Fig. 6. Structure of integrated I-MAKER and C-MAKER model with the

Figure 6 demonstrates the hierarchical structure of the integrated I-MAKER and C-MAKER model designed for pre-litigation liability decisions in the context of an employer (defendant) facing allegations from an employee (claimant). The individual legal pieces of evidence are processed by the I-MAKER approach. Subsequently, these pieces of evidence are combined by C-MAKER to yield a decision as a joint probability mass for each outcome within the power set of the frame of discernment (Θ) to incorporate uncertainty. For instance, the power set of possible decisions for "duty of care" is: $P(\Theta) = \{\{Duty\}, \{No\ Duty\}, \Theta = \{Duty, No\ Duty\}\}$, which can be represented simply as $\{Duty, No\ Duty, \Theta\}$. Similarly, the power sets for "breach of duty" and "causation" are represented as $\{Breach, No\ Breach, \Theta\}$ and $\{Established, Not\ Established, \Theta\}$, respectively. Three fundamental elements of employer liability are combined to provide a final liability decision for a given case.

The MAKER algorithm's ultimate decision on the acceptance or denial of liability relies on the accuracy of features representing the three principal elements of employer liability. The dataset had 3585 instances of past liability legal cases. Figure 7 illustrates the Area Under the Curve (AUC) and F1 scores from the 3-fold cross-validation sets for all fundamental elements of employer liability, along with the final liability decision.

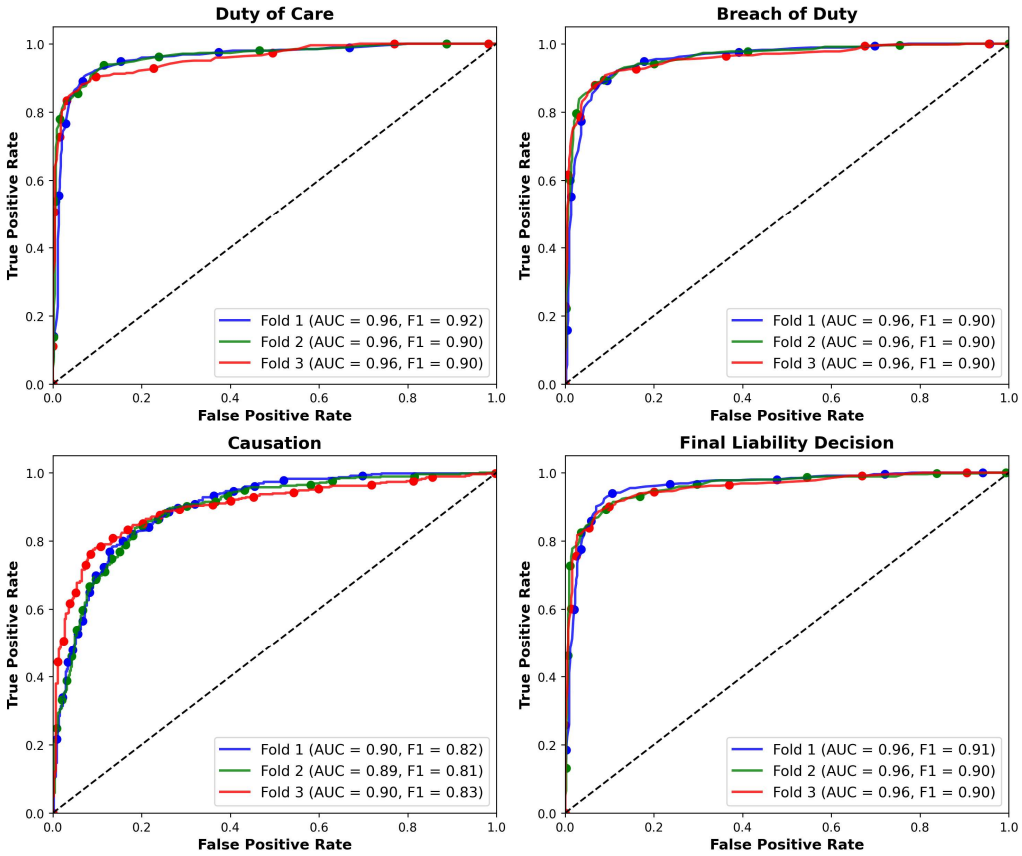

Fig. 7. ROC curve demonstrating AUC Score for employee liability decisions by integrated I-MAKER and C-MAKER model

The model intends to assist lawyers in comprehending the probable consequences of the legal facts of a case. Figure 8 and Figure 9 present the local explainability as the weight of the most relevant evidence for two distinct liability scenarios. These figures highlight a high joint probability mass towards the "Liability" decision for Case 1 and a slightly high joint probability mass for the "No-liability" decision for Case 2. These plots are generated from the data in a JSON file of a liability case, which is stored in an anonymized and encrypted format in the IPFS file system, with its hash value preserved in the blockchain for automated auditing.

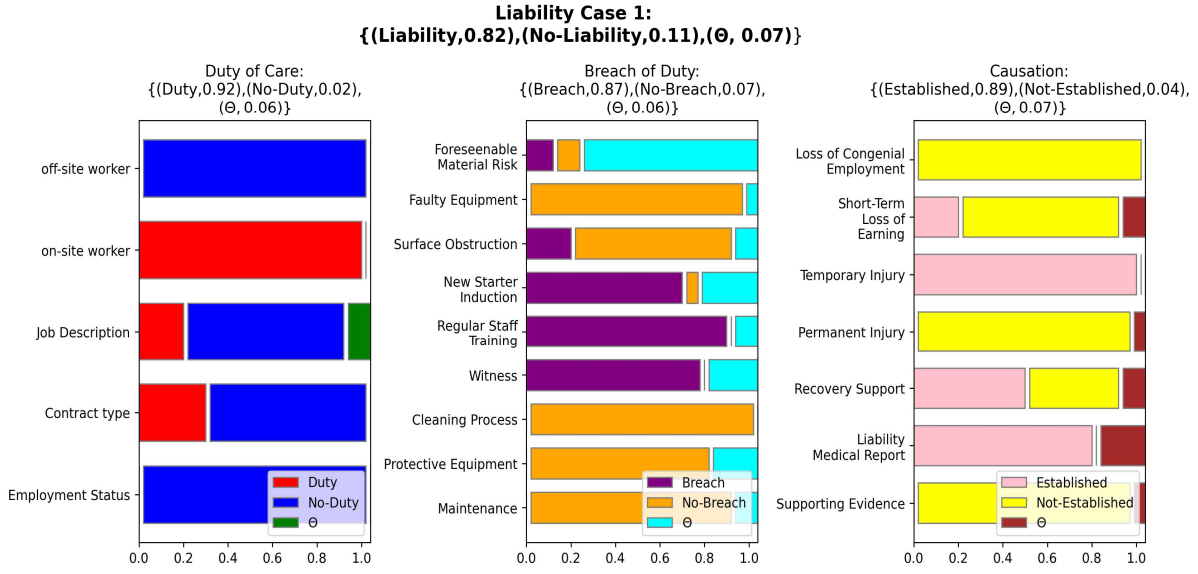

Fig. 8. Weight of most relevant evidence for an employee liability case, highlighting a high joint probability mass of "Liability" attributed to the employer negligence

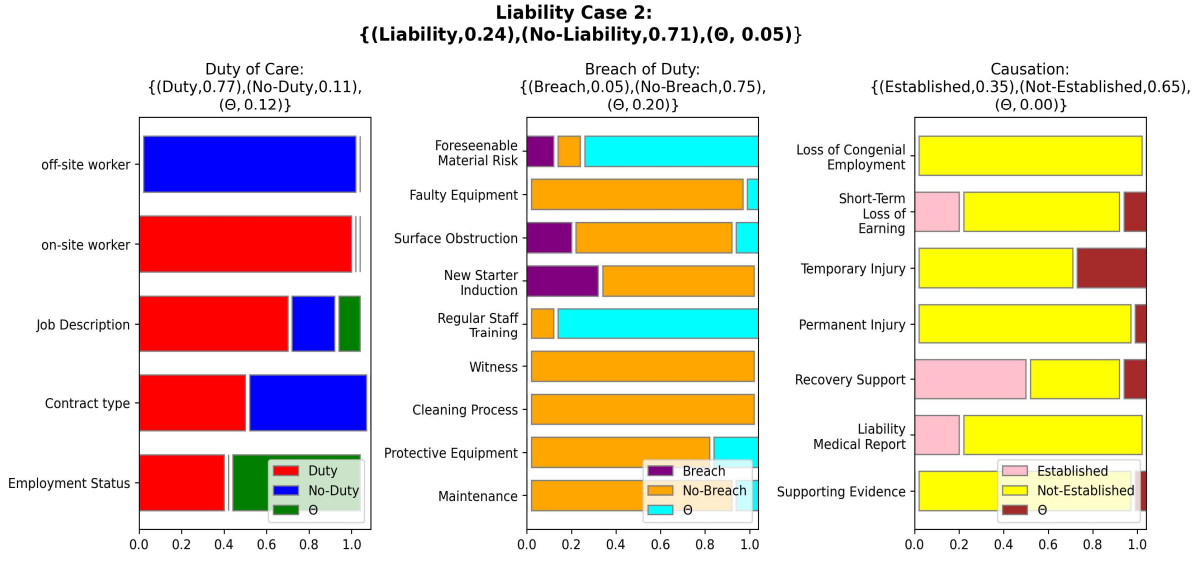

Fig. 9. Weight of most relevant evidence for an employee liability case, highlighting a minimal joint probability mass of "No-liability" attributed to the employer negligence

Figure 8 illustrates the weight of evidence examined in an employee liability case. It shows the ultimate decision of an employer's 'Liability' towards its employee. It emphasizes the interdependent relationship between the status of the employee - specifically an 'on-site worker' with valid 'employment status', 'contract type', and 'job description' - and the employer's responsibility to provide a safe working environment. In this case, the employer has breached his duty of care, primarily due to an inadequate 'new starter induction' and subpar 'Regular Staff Training'. This allegation is substantiated by compelling testimony from a 'witness', which further tilts the balance in the employee's favor. The link between the employer's breach of duty and the resulting injury to the employee is established through the 'Temporary Injury' sustained by the employee. The injury necessitated 'recovery support', as validated by the 'Liability Medical Report'. Therefore, the compiled evidence convincingly points towards a strong case for employer liability, highlighting the employer's failure to fulfill its duty of care, which resulted in significant harm to the employee.

On the contrary, Figure 9 presents a legal situation where the negligence of the employer could not be established. Despite the employer's duty of care towards its employee, there was insufficient evidential support to prove a breach of this duty or causation arising from the accident reported by the claimant (employee).

This section provided an introductory overview of the MAKER model's application to employer liability cases. It aimed to lay a foundation of understanding for the subsequent sections of the case study on the production and usability of legal content by Generative AI tools and the adoption of blockchain technology for auditing.

4.2 Generative AI for Textual Explanations of Legal Decisions by XAI

Multiple versions of large language models by OpenAI's models - GPT-3.5 ("text-davinci-003" and "text-davinci-002") and GPT-4 ("gpt-4"), and by Google Bard ("Bard") can be accessed by their respective APIs. Prompts were dispatched through a Python script in a loop to generate text for the legal decision rendered for each x^{th} defendant by an XAI algorithm. The decision by the XAI algorithm is stored and organized in a JSON format. In this case study, a defendant is an organization held responsible for the injury or loss suffered by its employee. Throughout this experiment, the structure of each input prompt transmitted through the loop by the Python script remains consistent as a string datatype. Figure 10 presents an instance of a prompt submitted for processing by the GPT-4 model, along with the resultant output. It presents a textual explanation of a legal decision obtained by the MAKER model for an employer liability case.

Prompt in Generative-AI Tool without Data Subject's Real Identifiers

```
Prompt: explain the output of following
liability claim by AI model for an
organisation
{
    "Defendant name": "XYZ Limited",
    "AI_Model": "XAI-Model",
    "ID_x": "DUMMY_ID_123",
    "Date Time": "YYYY-MM-DD HH:MM:SS",

    "Final Prediction":{
            "Liability": "0.41",
            "No-Liability":"0.59",
            "Uncertainty" = "0.0"},

    "Liability Predictions":{
            "Duty of Care": "No-Duty",
            "Breach of Duty":"No-Breach",
            "Causation":"Established"},

    "Weight":{
            "feature 1": "0.23",
            "feature 2":"1.00",....},

    "Relability":{
            "feature 1": "0.21",
            "feature 2: "1.0",...},
}
```

Content from Generative-AI Tool: Explanation of AI Model Decision

This JSON output represents the results of a liability claim analysis performed by our AI model, XAI-Model, for an organization named XYZ Limited. The AI model has provided its final predictions on various aspects of the liability claim.

The "Final Prediction" section indicates that the model calculates a 41% (or 0.41) probability of XYZ Limited being liable and a 59% (or 0.59) probability of them not being liable. The "Uncertainty" value of 0.0 implies the model is confident in its prediction.

Specifically, in terms of the key legal components of the claim, the AI model concluded the following:

1. "Duty of Care": The model determines that there is "No-Duty", meaning that it believes XYZ Limited does not owe a duty of care in this situation.

2. "Breach of Duty": The model also finds "No-Breach", indicating that it believes XYZ Limited has not violated any existing duty of care.

3. "Causation": However, the model states "Established", meaning there is a link between XYZ Limited's actions and the damage or harm in question.

Moreover, the output includes the weight and reliability metrics for each feature that the model considered during its analysis:

Weight: This section shows the importance or contribution of each feature to the model's prediction. For instance, 'feature 1' has a weight of 0.23, while 'feature 2' has a weight of 1.00. The higher the weight, the more impactful the feature is on the model's decision.

Reliability: This part provides the reliability scores of each feature used by the model. The scores range from 0 to 1, with higher scores representing greater reliability. In this example, 'feature 1' has a reliability score of 0.21, while 'feature 2' has a score of 1.0.

In conclusion, the AI model proposes that XYZ Limited is more likely to be absolved of liability in this particular claim, based on the lack of a duty of care, absence of breach of any duty, and established causation. This assessment is reflected in the Liability and No-Liability probabilities, as well as the weights and reliability scores of the individual features considered during the analysis. However, it's essential to remember that this conclusion doesn't replace a thorough legal examination.

Fig. 10. An example of a prompt for the GPT-4 model and the corresponding response based on the explanation provided by the integrated I-MAKER and C-MAKER model

4.2.1 Text Usability Evaluation

An experimental study was conducted to assess the usability of text generated by Google Bard and three OpenAI models. In this investigation, 50 final-year law school students were selected to participate in the task of drafting legal decisions on employer liability cases. The students were divided into two groups - an experimental and a control group, each comprising 25 students. The students in the experimental group utilized automated textual explanations of legal decisions by Generative AI tools to draft liability decision letters. In contrast, the control group students composed their decision letters without the aid of Generative AI. The information on a general summary of the study's characteristics is shown in Table 7.

Table 7. Characteristics of Experimental Study

Study Aspect	Detail
Participants	50 final-year law students
Group Division	Experimental group (25 students with Generative AI assistance); Control group (25 students without Generative AI assistance)
Assessors	faculty of law school (Lecturers)
Task	Drafting liability decision letters
Measurement Tool	Turnitin plagiarism software (used to measure the utilization score)

Generative AI Models Evaluated	Google Bard: “Bard”, GPT-4: “gpt-4”, GPT 3.5: “text-davinci-003”, “text-davinci-002”

The law school faculty assessed the quality of the letters drafted by the students. Each letter was graded on a scale from 0 to 100. The Turnitin plagiarism software was employed to determine the extent of textual similarity, termed the utilization score, between student-composed drafts and the text generated by OpenAI and Google Bard algorithms. The draft composed by the control and experimental group students, along with text generated by the AI tools, was uploaded to the university course submission page.

Turnitin was chosen for this task due to its proficiency in assessing the level of similarity between the examined work and other written pieces (Manley, 2023). The experiment excluded the similarity percentages derived from external web sources to focus solely on the AI-generated texts. The results in Figure 11 show that the GPT-4 model excelled in generating high-quality content, with 58.5% of its output being utilized to draft legal documents. The "Bard” utilization score was inferior to “gpt-4”. Conversely, the law students opted to use 31.3% and 19.5% of the content produced by the “text-davinci-003” and “text-davinci-002” models, respectively.

In terms of drafting time, drafting a letter using the “text-davinci-002” model took a slightly longer average time than the control group. However, the “gpt-4” model expedited the drafting process, with an average letter completion time of just 6 minutes. It outperformed "Bard", which required an average of 8 minutes. The grades assigned by the law faculty for the letters produced by the control group and those utilizing the "Bard" and "gpt-4" models fell within a similar range of 73 to 75. This indicates that the quality of the letters produced across these groups maintained a consistent standard, irrespective of the model. However, the time required to draft a letter varies depending on the generative AI tool employed. For instance, "gpt-4" stands out as an exemplary tool that generates high-quality content and shortens the time needed to draft the letter. On the other hand, "text-davinci-002" presents a contrasting scenario; its output is of lesser quality and demands a longer drafting period.

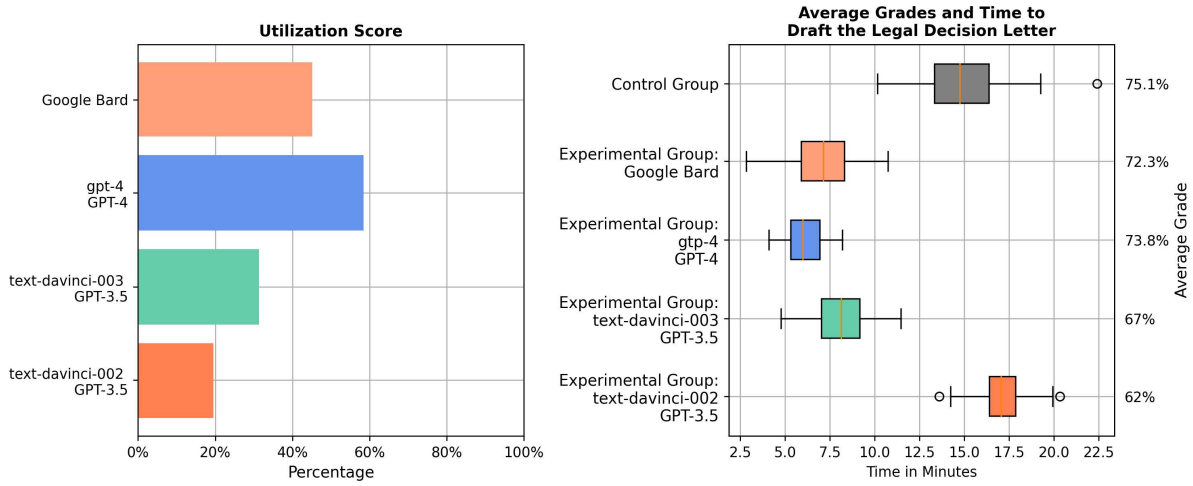

Fig. 11. Usability test of generative AI models

4.3 Experimental Results on Auditing by Blockchain

A comparative experimental evaluation of the proposed framework is conducted on two distinct blockchain networks: Ethereum 2.0 and Hyperledger Fabric 2.0. Ethereum can support a public permissionless Mainnet or a private network, known for its principle of anonymity. Hyperledger Fabric is designed for private consortiums where identity authentication and access rights validation are critical. Ethereum provides a mass audience and is more decentralized than Hyperledger Fabric. The assessment does not lean towards any network type. It provides a balanced examination of the framework's capabilities across different blockchain environments.

***Experimental Setup*:** Table 8 (a) demonstrates the test characteristics of the environment designed to evaluate the proposed blockchain-based auditing framework. Two identical machines hosted distinct blockchain nodes: the hash storage and audit nodes. Both machines operated on Ubuntu 19.04 with an 8-core CPU, 16 GB RAM, and 700 GB of available disk space. The complete copy of the ledger data of Ethereum was stored in LevelDB and Hyperledger Fabric in CloudDB.

Table 8 (b) demonstrates technology specification. Microsoft Azure Key Vault was employed for the secured storage of encryption keys, node IDs, and digital signatures for an added layer of security. The encrypted and anonymized metadata derived from the XAI algorithm and Generative AI tools were stored within Pinata (IPFS service provider). The data in IPFS is stored as a copy of data in the Cloud. Azure Blob Cloud storage was utilized to store the encrypted but deanonymized legal information to maintain centralized control.

Test Aim: Multiple users access the data stored in the Cloud and IPFS in an organization, which increases the susceptibility to intentional or unintentional tampering. In response to this vulnerability, a blockchain-based auditing system is implemented to verify the integrity of the data stored in the Cloud and IPFS.

Table 8 (a): Environment Specifications

Component	**Environment**
Test Machines	Operating System: Ubuntu 19.04
Hardware	8-Core CPU, 16 GB RAM, 700 GB Disk Space
Ledger Data Storage	Ethereum: LevelDB, Hyperledger Fabric: CouchDB
API	FastAPI, a high-performance web framework for building REST APIs
API Server	Postman to get and send response to API endpoints
Broadband Internet Speed	433Mbps to 1.3Gbps

Table 8 (b): Technology Specifications

Technology	**Service Provider**
Blockchain Network (On-chain Data Storage)	**Hyperledger Fabric (Version 2.0):** Permissioned, private network
	Ethereum (Version 2.0): Permissionless, public network
Cryptographic Key Management	**Azure Key Vault:** A Cloud service provided by Microsoft Azure for secure storage and management of cryptographic keys, node IDs, and digital signatures.

Off-chain Data Storage	**IPFS by Pinata:** Pinata is an IPFS pinning service provider. It is utilized to store the encrypted and anonymized metadata of the XAI algorithm and content produced by Generated AI tools.
	Azure Blob Storage: Centralized, secure storage of legal data

The blockchain network performance is assessed by throughput, latency, and resource consumption (Kuzlu, et al., 2019) (Pongnumkul, et al., 2017). Throughput is the number of valid transactions the blockchain commits per unit of time (usually in seconds). It is also called the transaction per second (TPS) rate. Latency is the time required for a transaction to be confirmed, or in other words, the time a blockchain network takes to broadcast a transaction once it is dispatched to a node administered by an organization.

The increase in throughput decreases the latency. The trade-off between throughput and latency is uneven. Latency goes up sharply when the load to record data approaches its maximum throughput. We managed this by utilizing the batch operation approach that sends multiple transactions within a block. Utilizing a single block to register a transaction associated with a specific legal case could potentially lead to a decrease in throughput and a corresponding rise in transaction costs because there is a fixed transactional cost involved in recording data in a single block of the blockchain. A batch operation waits for the arrival of multiple transactions before it is released in the network to get recorded in a block. Sending multiple transactions as a batch within a single block could significantly improve throughput and save transaction fees.

Table 9: Network Parameters to Test Performance

Hyperledger Fabric Parameters	**Parametric Setting**	
	Scenario 1	**Scenario 2**
Batch Timeout: This is the wait time after the first transaction's arrival during which additional transactions can be added before a block is finalized.	5 seconds	10 seconds
Max Message Count: The upper limit for the permitted number of transactions in a batch allowed to be recorded in a block.	5	10
Absolute Max Bytes: This is the maximum limit on the number of bytes permitted for serialized messages within a batch. Any transactions that exceed this size are rejected.	10 MB	10 MB
Preferred Max Bytes: It is the optimal maximum size of a batch, represented as the maximum number of bytes allowed for serialized messages within it. A batch continues to fill with transactions until this size, the max message count, or the batch timeout is reached. If adding a new message (or transactions) causes the batch to exceed the preferred max bytes, the current batch is then finalized, committed to a block, and a new batch is initiated with the new message. If a single message exceeds the preferred max bytes, it is assigned its own unique batch.	512 KB	512 KB
Ethereum Parameters	**Parameter Value**	

Gas Limit: The maximum amount of Gas consumed by sending and executing a transaction by an Ethereum Virtual Machine that operates the nodes.	100,000	100,000
Gas Used by Transaction: The estimated amount of Gas required to execute a transaction.	40,000	80,000

(a) Performance

The performance of the proposed blockchain-based audit framework was evaluated using two different parametric settings of Hyperledger Fabric and the Ethereum network. The average size of on-chain legal data, which includes hash and Merkle root recorded on the blockchain, ranged between 0.01 MB and 0.25 MB. The performance of Hyperledger was analyzed by adjusting the parameters on batch timeout, max message count, absolute max bytes, and preferred max bytes. These parameters determine the number of transactions in a batch and the time required to broadcast a batch inside a block of the Hyperledger network. The batch timeout denotes the interval following the arrival of the initial legal case after the prior block broadcast, during which additional independent cases can be gathered. These accumulated cases are then recorded collectively as a batch within a single block.

In Hyperledger Fabric, the default preferred maximum bytes are 512 KB, and an absolute maximum byte is set at 10 MB. We experimented on Hyperledger Fabric in two separate batch timeouts: 5 and 10 seconds and maximum message counts: 5 and 10. This implies that a block can accommodate a batch of 5 legal cases with a batch timeout of 5 seconds or a batch of 10 legal cases with a batch timeout of 10 seconds.

The number of legal cases to be recorded as a batch in the Ethereum network depends on the max amount of gas consumed by sending and receiving the data in a block and the amount of gas required to execute a transaction for an individual legal case successfully. The maximum Gas limit in Ethereum is set to be 100,000. This limit is the maximum amount the user is willing to pay in the event of gas price fluctuations. We tested the proposed framework with two different Gas values per Ethereum transaction: 40,000 and 80,000. Table 9 defines these parameters and their respective test values.

Figure 12 illustrates the throughput and latency of Hyperledger and Ethereum while varying the number of transactions from 1 to 10,000 under the two distinct parametric configurations. It can be observed that the increase in batch timeout from 5 secs to 10 secs and maximum message count from 5 to 10 improved the throughput of the Hyperledger Fabric. Similarly, the increase in gas cost for a transaction from 40,000 to 80,000 improved the throughput of Ethereum. However, both networks experienced an increase in network latency with expanded capacity to handle larger batches. The observed latency is marginal compared to the volume of legal cases a law firm handles in a typical working day. This suggests that the firm can prioritize throughput and cost savings without compromising overall efficiency.

Hyperledger Fabric demonstrated better performance compared to Ethereum. It exhibits higher throughput and reduced latency in both parametric settings. This performance gap becomes increasingly noticeable with the increase in the number of transactions.

Both the hash storage node and the audit node keep their individual copies of data on the blockchain network. It augments the computational and storage scalability of the blockchain application, as no single node is overwhelmed with the responsibility of managing the entire network's transactions. The data partitioning across multiple nodes enhances data security.

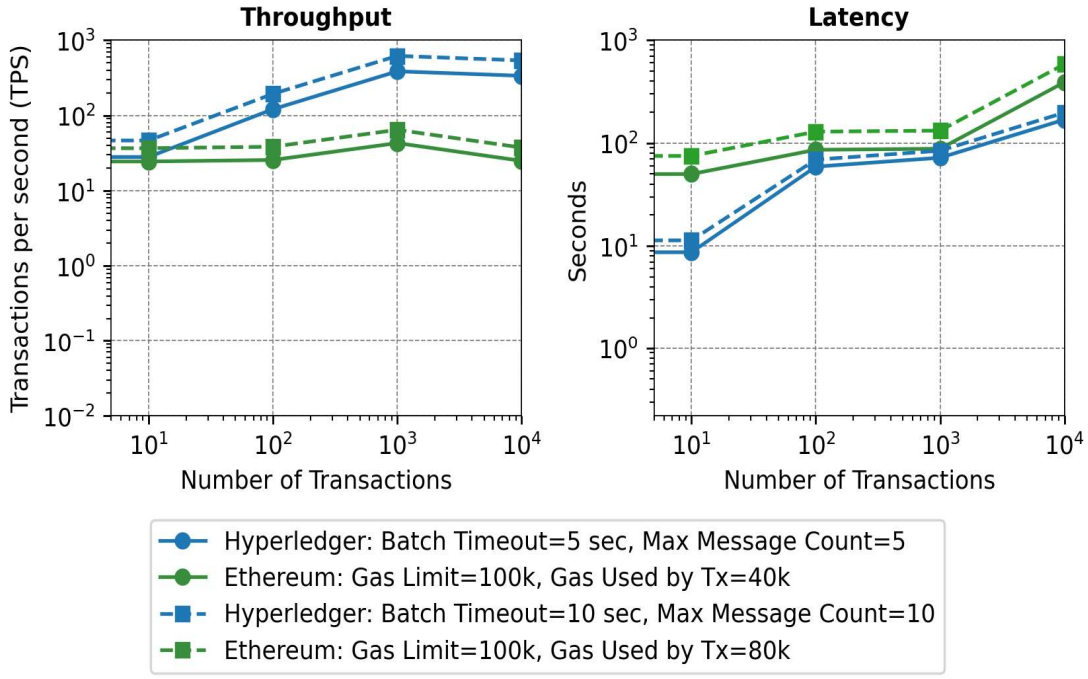

Fig. 12. Performance of Ethereum and Hyperledger Fabric: Latency and Throughput

The volume of requests for updating information in a blockchain platform tends to grow concurrently with the increase in the number of users and nodes (servers). We further assessed the scalability of Ethereum and Hyperledger by systematically increasing the number of nodes from 4 to 28. The transactions (network traffic) were distributed evenly across all nodes.

The parameters for the scalability test were maintained consistent with scenario 1, discussed in the previous section. This controlled setting provided a clear comparison between the performance metrics of Ethereum and Hyperledger, highlighting their capacity to manage escalated loads with the growth in the number of servers. Figure 13 illustrates the resulting throughput and latency as we scaled the workload by increasing the number of nodes. In both networks, throughput decreased, and latency increased beyond 12 nodes. Moreover, Hyperledger Fabric failed to process all transactions beyond 20 nodes, indicating a potential limitation in its scalability under these settings.

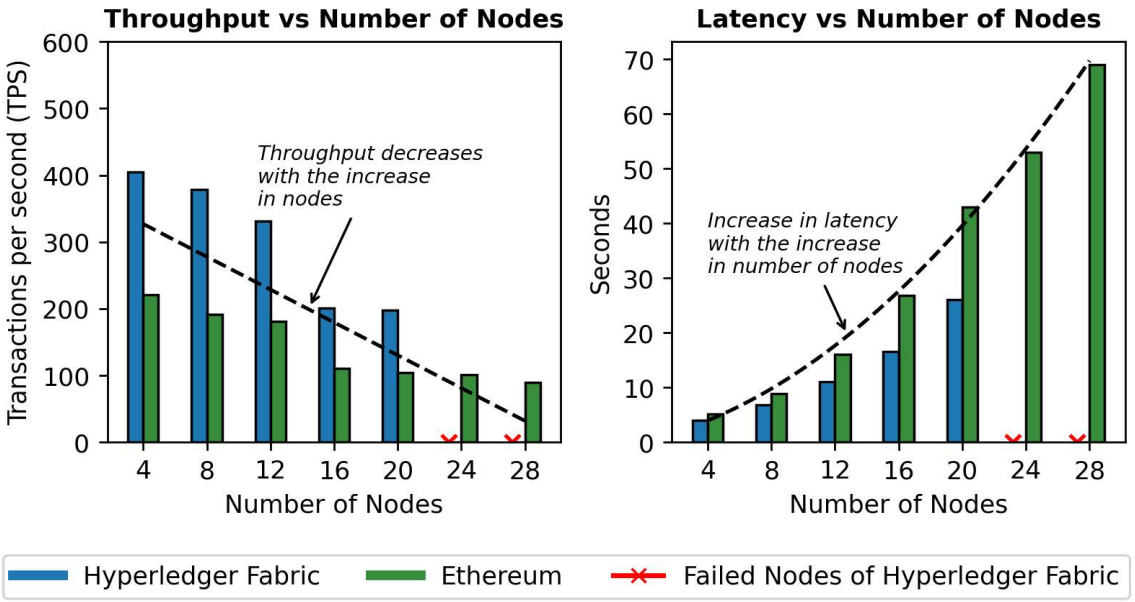

Fig. 13. Latency and Throughput in response to the node scaling

(b) Memory and CPU Consumption

The smart contract in Ethereum and the chain code in Hyperledger Fabric carry out two fundamental tasks. The 'Open' function is responsible for creating an account within the participating nodes, and the 'Query' function probes the peer ledger for the extraction of on-chain data to audit off-chain files. Figure 14 illustrates the average memory consumption and CPU usage of both functions across both platforms. There is a significant disparity in memory consumption between both networks. Ethereum consumes nearly seven times more memory than Hyperledger Fabric. Similarly, the average computational power used by Ethereum's CPU is comparatively higher than that of Hyperledger Fabric.

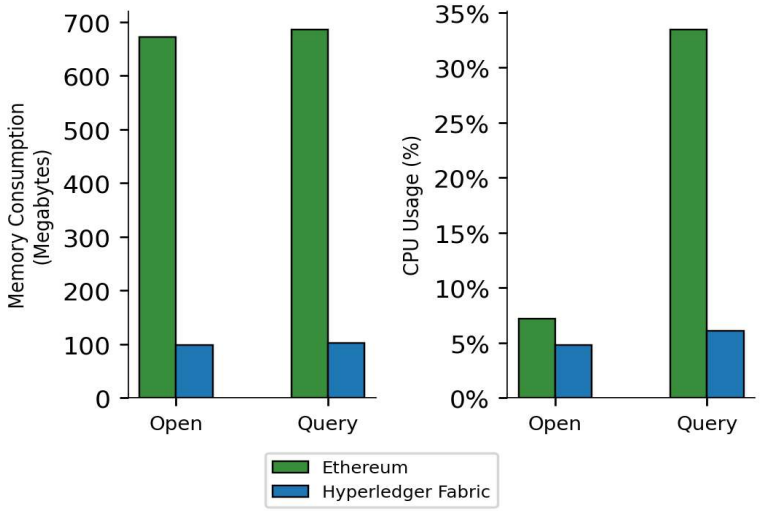

Fig. 14. Average memory consumption and CPU usage

(c) Auditing to Detect Tampered Files

The efficiency of our audit methodology was evaluated by introducing random manipulations in 2% to 16% of legal case files stored in off-chain mediums. To pass auditing the Merkle root of off-chain storage mediums: IPFS and Cloud must match with their respective Merkle root stored permanently in a blockchain platform (on-chain). The IPFS possesses a high level of security because any change in IPFS files will result in a change to its CID. IPFS acts as a secure storage for large files unsuitable for on-chain storage and a supplementary backup to Cloud storage. However, these off-chain files are frequently accessed by multiple organizational users, which increases the likelihood of intentional or unintentional tampering. The widespread accessibility heightens the system's vulnerability to the tampering of files by internal or external malicious actors.

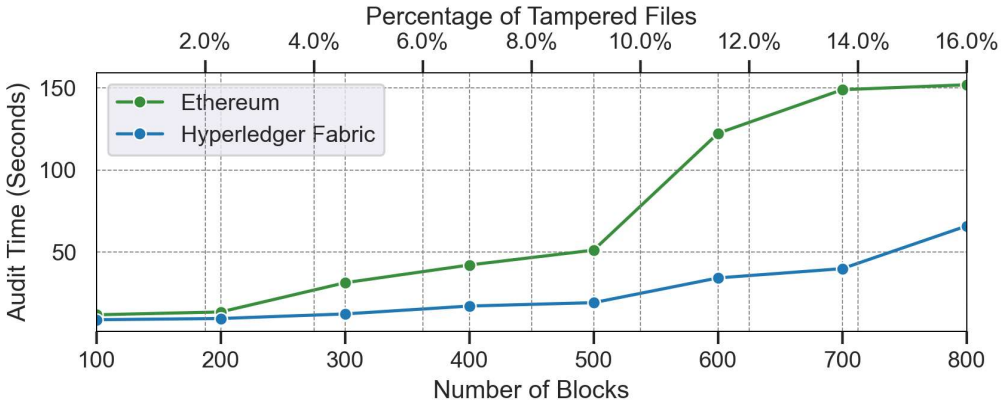

Fig. 15. Auditing time of data block

The rigorous auditing process ensures that all files remain consistent from the moment a legal case was processed in the past, and the case files were securely stored in both off-chain and on-chain repositories. Automated blockchain-based auditing verifies the integrity of both off-chain and on-chain data files. It can effectively eliminate any inconsistencies in tracing the accountability of adversarial legal decisions. Tampering and questionable decisions indicate deviations from ethical standards.

The duration of the auditing process escalates with the increase in the proportion of tampered files, as shown in Figure 15. The hashed identifier corresponding to each detected tampered file is recorded within the hash node as a permanent, unalterable audit report. The frequency of the blockchain-based automated auditing activity can be adapted to the firm's needs, varying from weekly reviews to numerous audits within a week.

(d) Comparative Analysis

The comparative analysis benchmarks the proposed blockchain-based audit technique against two similar methods for performance evaluation. First is the Blockchain Audit Trail (BAT) algorithm, which utilizes Merkle Tree authentication to verify data integrity in monitoring forest fires (Datta & Sinha, 2023). Second, auditing of XAI decisions by storing the hash of an IPFS file on the Ethereum blockchain to avoid large file storage in the blockchain (Malhotra, et al., 2021). The proposed method under scrutiny in this study utilizes Merkle root for audit trails and IPFS to store large off-chain legal files in an anonymized and encrypted format.

The comparative analysis was conducted on Hyperledger Fabric with the parameters set out in Scenario 1, as illustrated in Table 9. The results show that the proposed technique exhibits a higher throughput and

marginally less audit time. It outperforms Method 1, based on the Merkle root authentication approach by Datta & Sinha (2023), and Method 2, the use of IPFS and blockchain for auditing XAI decisions by Malhotra et al. (2021), as shown in Figure 16. The high throughput of the proposed approach in this research can be attributable to batch processing, which accumulates transactions for five seconds or until five transactions are queued, whichever comes first, before broadcasting the batch to the network. Additionally, the architecture separates responsibilities across two distinct nodes: one dedicated to hash storage and the other to auditing tasks to balance the computational and storage demands. Overall, the proposed method achieved a 23.65% and 21.67% higher throughput than Method 1 and Method 2, respectively. Similarly, the audit time of the proposed method was 16.36% and 9.38% lower than Method 1 and Method 2, respectively.

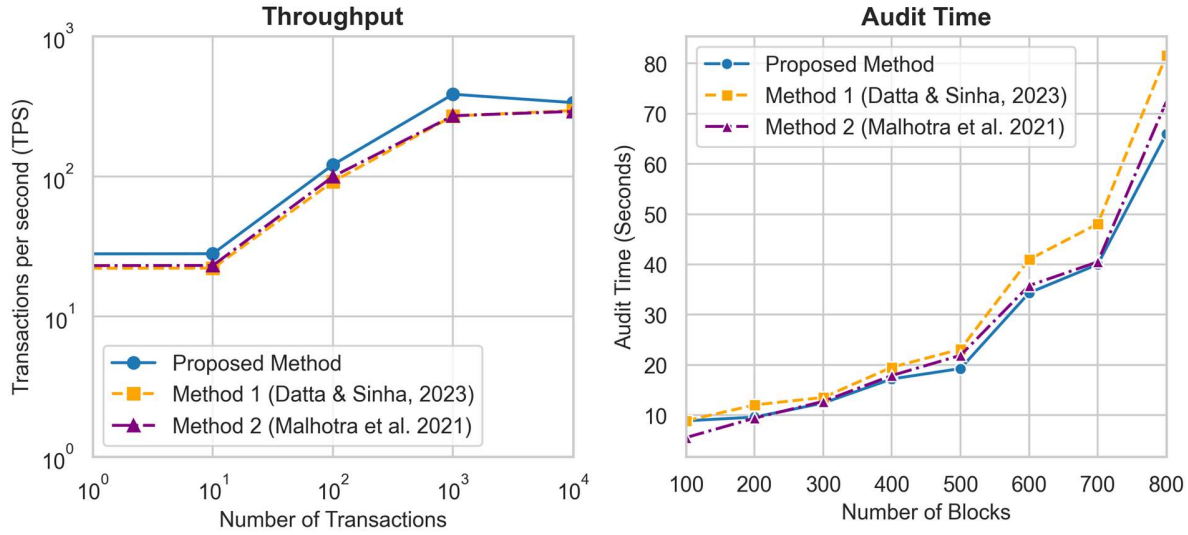

Fig. 16. Comparative results on throughput and auditing time

5. Limitations and Future Improvements

In this research, the nodes were self-deployed and self-managed by machines that met the minimum hardware requirements essential for the successful deployment of the nodes. Operating physical nodes in companies requires expertise in hardware management, an unrestricted broadband connection, and robust onsite security measures to safeguard the equipment. However, blockchain service providers resolve technical issues and manage all the necessary tasks and activities to keep the blockchain infrastructure operational. In the future, we intend to leverage the services of specialized node providers who offer access to the nodes.

This paper advances the formulation of evidential reasoning (ER) for legal knowledge representation to process ambiguous legal facts. Legal syllogistic reasoning typically employs Bayesian methods. However, ER is based on Dempster–Shafer's theory of evidence, an extension of Bayesian theory. We acknowledge that the automation of legal decisions through AI has limitations due to the complex and dynamic nature of law. We propose the future development of a 'Lawyer-in-the-Loop' augmentation framework to facilitate an active knowledge exchange between legal practitioners and AI systems. Advanced techniques such as knowledge distillation (Xiao, et al., 2023) can be incorporated to transfer insight from complex, high-level legal concepts to more specific, low-level legal concepts. The ER approach is demonstrated in tort law. It can be adapted to diverse branches of law, such as constitutional law, criminal law, corporate law, intellectual property law, and contract law.

6. Conclusion

We explored the promising potential of Generative AI tools in brainstorming ideas to draft correspondence for pre-litigation decisions. This research has presented an innovative end-to-end process for designing architecture and methodology for a blockchain-based auditing system. The system maintains the integrity of data repositories containing the decisions by an XAI model and textual explanation of legal cases obtained through a prompt sent to the API of Generative AI. Automated auditing enhances data security, enforces responsible usage of AI technologies, and reduces discrepancies in tracing the accountability of adversarial legal decisions. The system utilizes the immutable nature of blockchain – a feature that is invaluable in auditing but may fall short in data protection laws. The proposed system's architecture aligns with critical data protection laws.

The research conceptualizes the blockchain-based auditing process through two algorithms. The first algorithm presents a comprehensive strategy for on-chain and off-chain storage of legal data. The second algorithm outlines the auditing process based on the principle of comparing locally recomputed Merkle roots of files stored in off-chain storage with their corresponding immutable Merkle root stored in a blockchain network. The security analysis validates the robustness of the blockchain-based auditing system's potential to resist malicious actors' attempts to manipulate on-chain data to bypass the auditing process.

The paper advances the formulation of an integrated I-MAKER & C-MAKER model to methodically capture ambiguous legal facts and heuristics to establish a causal relationship between evidence and the final hypothesis (or decision). The explainable algorithmic decisions from this model are anonymized to safeguard sensitive legal information by substituting the data subject's identifiable information with a dummy placeholder to ensure the confidentiality and privacy of prompts sent to the APIs of Generative AI tools. The drafted correspondence for legal decisions was generated through APIs that grant access to LLM algorithms by Generative AI tools. The study tested four distinct LLMs: Google's Bard ("Bard"), GPT-4 ("gpt-4"), and GPT 3.5 ("text-davinci-003" and "text-davinci-002"). Results from an experimental study at a law school revealed the GPT-4 model's proficiency in producing high-quality content and significantly speeding up the drafting process compared to other models. The practical implementation of the proposed framework is demonstrated through a case study based on liability cases arising from workplace accidents. The proposed technique can be adapted to tort claims such as economic, property, dignitary, strict & absolute liability, and nuisance.

The blockchain-audit algorithm was tested to audit the on-chain files containing the decisions derived from the XAI algorithm based on MAKER and its textual explanations produced by Generative AI. This assessment was conducted on two distinct blockchain platforms: Ethereum, a public platform, and Hyperledger Fabric, a private platform. The off-chain data (outside blockchain) was stored within IPFS files facilitated by Pinata, an IPFS service provider, and secured Cloud by Azure Blob. Microsoft Azure Key Vault was utilized for the secure storage of encryption keys, node IDs, and digital signatures. The integrated use of multiple technologies to test the proposed system highlights the inherent complexity of a blockchain application, which leans on the support and collaboration of technology service providers for optimal functionality and security.

This research illustrates the profound implications of integrating cutting-edge technologies in the legal industry, particularly in fostering responsible usage of AI-generated content and AI-assisted decision-making for lawyers. The findings support the potential for widespread adoption of these technologies to pave the way for a more transparent and reliable future in the legal industry. The blockchain-based data

integrity audit methodology presented in this study can be extended to other domains. Organizations interested in monitoring Generative AI usage can adapt the proposed framework to fit their specific requirements.

Acknowledgment

We are grateful to the four anonymous reviewers for their insightful comments, which significantly improved our manuscript. This work is fully funded by the University of Liverpool by the research grant 106897. We want to express gratitude to the law students who participated in our experimental study; their contributions were invaluable.; their contributions were invaluable. The first author is grateful for the substantive knowledge exchange during the AI Claims Handler project focused on liability claim handling, funded by UKRI. The first author is thankful to Prof Walter Davis Jr. and Mr. Graham Fairclough for valuable advice on the integration of AI and Blockchain in legal practices.

Bibliography

Agyekum, K. O. B. O. et al., 2021. A proxy re-encryption approach to secure data sharing in the internet of things based on blockchain. *IEEE Systems Journal ,* pp. 1685-1696.

Ajevski, M. et al., 2023. ChatGPT and the future of legal education and practice. *The Law Teacher ,* pp. 1-13.

Al-Abdulkarim, L., Atkinson, K. & Bench-Capon, T., 2014. Abstract dialectical frameworks for legal reasoning. *In Legal Knowledge and Information Systems ,* pp. 61-70.

Aleven, V. A., 1997. Teaching case-based argumentation through a model and examples. *Pittsburgh: University of Pittsburgh.*

Bell, J., 2013. The basis of vicarious liability. *The Cambridge Law Journal,* Volume 72(1), pp. 17-20.

Bendiab, G. et al., 2023. Autonomous vehicles security: Challenges and solutions using blockchain and artificial intelligence. *IEEE Transactions on Intelligent Transportation Systems.*

Biever, C., 2023. ChatGPT broke the Turing test-the race is on for new ways to assess AI. *Nature,* pp. 619(7971), 686-689.

Binder, A. et al., 2016. Layer-wise relevance propagation for deep neural network architectures. *Information science and applications (ICISA), Springer Singapore,* pp. 913-922.

Bruninghaus, S. & Ashley, K. D., 2003. Predicting outcomes of case based legal arguments. *In Proceedings of the 9th international conference on Artificial intelligence and law,* pp. 233-242.

Bushard, B., 2023. Workers' ChatGPT Use Restricted At More Banks—Including Goldman, Citigroup. *Forbes.*

Choi, J. H., Hickman, K. E., Monahan, A. & Schwarcz, D., 2023. Chatgpt goes to law school. *Available at SSRN.*

Collenette, J., Atkinson, K. & Bench-Capon, T., 2023. Explainable AI Tools for Legal Reasoning about Cases: A Study on The European Court of Human Rights. *Artificial Intelligence,* p. 103861.

Collins, E. & Ghahramani, Z., 2021. LaMDA: our breakthrough conversation technology. *Google AI Blog.*

Conrad, E., Misenar, S. & Feldman, J., 2016. Chapter 4—Domain 3: Security Engineering (Engineering and Management of Security). *CISSP Study Guide, 3rd ed.; Syngress: Boston, MA, USA,* pp. 103-217.

Constant, A., 2023. A Bayesian model of legal syllogistic reasoning. *Artificial Intelligence and Law,* pp. 1-22.

Das, M., Tao, X. & Cheng, J. C., 2021. BIM security: A critical review and recommendations using encryption strategy and blockchain. *Automation in construction,* p. 103682.
Datta, S. & Sinha, D., 2023. BSEIFFS: Blockchain-secured edge-intelligent forest fire surveillance. *Future Generation Computer Systems,* pp. 59-76.
Davies, M., 1989. The end of the affair: duty of care and liability insurance. *Legal Studies,* Volume 9(1), pp. 67-83.
Dempster, A., 2008. *Upper and lower probabilities induced by a multivalued mapping.* Berlin Heidelberg, Springer, pp. 57-72.
Du, Y. W. & Zhong, J. J., 2021. Generalized combination rule for evidential reasoning approach and Dempster–Shafer theory of evidence. *Information Sciences,* Volume 547, pp. 1201-1232.
Dwivedi, Y. et al., 2023. So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy.. *International Journal of Information Management,* p. 102642.
Fenton, N., Neil, M. & Berger, D., 2016. Bayes and the law. *Annual review of statistics and its application,* pp. 3, 51-77.
Finck, M., 2019. Blockchain and the General Data Protection Regulation: can distributed ledgers be squared with European data protection law?: study. *European Parliament.*
Fu, C., Yang, J. B. & Yang, S. L., 2015. A group evidential reasoning approach based on expert reliability. *European Journal of Operational Research,* pp. 886-893.
Galanti, R. et al., 2023. An explainable decision support system for predictive process analytics. *Engineering Applications of Artificial Intelligence,* p. 105904.
GDPR Article 5, 2023. A guide to the data protection principles. *https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/data-protection-principles/a-guide-to-the-data-protection-principles/the-principles/data-minimisation/.*
Han, H. et al., 2023. Accounting and auditing with blockchain technology and artificial Intelligence: A literature review. *International Journal of Accounting Information Systems,* p. 100598.
Hao, K., Xin, J., Wang, Z. & Wang, G., 2020. Outsourced data integrity verification based on blockchain in untrusted environment. *World Wide Web,* Volume 23, pp. 2215-2238.
Hariharasitaraman, S. & Balakannan, S. P., 2019. A dynamic data security mechanism based on position aware Merkle tree for health rehabilitation services over cloud. *Journal of Ambient Intelligence and Humanized Computing,* pp. 1-15.
Hasan, H. R. et al., 2020. Blockchain-based solution for COVID-19 digital medical passports and immunity certificates. *IEEE ACCESS,* pp. 222093-222108.
Hasan, H. R. et al., 2020. Blockchain architectures for physical internet: A vision, features, requirements, and applications. *IEEE Network,* pp. 174-181.
Huang, Q., Quan, L. & Zhang, S., 2022. Downsampling and transparent coding for blockchain. *IEEE Transactions on Network Science and Engineering,* pp. 2139-2149.
Iu, K. Y. & Wong, V. M. Y., 2023. ChatGPT by OpenAI: The End of Litigation Lawyers?. *Available at SSRN.*
Joe, C. V. & Raj, J. S., 2021. Deniable authentication encryption for privacy protection using blockchain. *Artificial Intelligence and Capsule Networks,* pp. 259-271.
Juels, A. & Kaliski Jr, B. S., 2007. PORs: Proofs of retrievability for large files. *Proceedings of the 14th ACM conference on Computer and communications security,* pp. 584-597.
Kang, P., Yang, W. & Zheng, J., 2022. Blockchain Private File Storage-Sharing Method Based on IPFS. *Sensors,* pp. 22(14), 5100.

Karisma, K. & Moslemzadeh Tehrani, P., 2023. *Blockchain: Legal and Regulatory Issues.* s.l., Cham: Springer International Publishing, pp. 75-118.
Katz, D. M., Bommarito, M. J., Gao, S. & Arredondo, P., 2023. Gpt-4 passes the bar exam. *Available at SSRN 4389233.*
Kushwaha, S. S. et al., 2022. Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access,* pp. 6605-6621.
Kuzlu, M., Pipattanasomporn, M., Gurses, L. & Rahman, S., 2019. Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. *2019 IEEE international conference on blockchain (Blockchain),* pp. 536-540.
Liu, B. et al., 2017. Blockchain based data integrity service framework for IoT data. *IEEE International Conference on Web Services (ICWS),* pp. 468-475.
Liu, B. et al., 2017. Blockchain based data integrity service framework for IoT data. *IEEE International Conference on Web Services (ICWS) ,* pp. 468-475.
Liu, X., Sachan, S., Yang, J. B. & Xu, D. L., 2019. Maximum likelihood evidential reasoning-based hierarchical inference with incomplete data. *25th IEEE International Conference on Automation and Computing,* pp. 1-6.
Lundberg, S. M. & Lee, S. I., 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems,* Volume 30.
Malhotra, D., Srivastava, S., Saini, P. & Singh, A. K., 2021 . Blockchain based audit trailing of XAI decisions: Storing on IPFS and Ethereum Blockchain. *IEEE International Conference on COMmunication Systems & NETworkS (COMSNETS),* pp. 1-5.
Manley, S., 2023. The use of text-matching software's similarity scores. *Accountability in Research,* pp. 219-245.
Manzoor, A. et al., 2021. Proxy re-encryption enabled secure and anonymous IoT data sharing platform based on blockchain. *Journal of Network and Computer Applications,* p. 102917.
Martino, R. & Cilardo, A., 2020. Designing a SHA-256 processor for blockchain-based IoT applications. *Internet of Things,* pp. 11, 100254.
Montavon, G. et al., 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition,* Volume 65, pp. 211-222.
Morris, C., 1952. Duty Negligence and Causation. *U. Pa. L. Rev.,* p. 101.
Muralidharan, S. & Ko, H., 2019. An InterPlanetary file system (IPFS) based IoT framework. *IEEE international conference on consumer electronics (ICCE),* pp. 1-2.
Nassar, M., Salah, K., ur Rehman, M. H. & Svetinovic, D., 2020. Blockchain for explainable and trustworthy artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* p. e1340.
Neil, M., Fenton, N., Lagnado, D. & Gill, R., 2019. Modelling competing legal arguments using Bayesian model comparison and averaging. *Artificial intelligence and law,* pp. 27, 403-430.
Nepal, S., Chen, S., Yao, J. & Thilakanathan, D., 2011. DIaaS: Data integrity as a service in the cloud. *IEEE 4th International Conference on Cloud Computing,* pp. 308-315.
Nizamuddin, N., Hasan, H. R. & Salah, K., 2018. IPFS-blockchain-based authenticity of online publications. *In Blockchain–ICBC 2018: First International Conference, Held as Part of the Services Conference Federation, SCF 2018, Seattle, WA, USA, June 25-30, 2018, Proceedings 1,* pp. 199-212.
Norkute, M. et al., 2021. Towards explainable AI: Assessing the usefulness and impact of added explainability features in legal document summarization. *In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems,* pp. 1-7.
OpenAI, 2023. GPT-4 Technical Report.

Philip, A. O. & Saravanaguru, R. K., 2023. Multisource traffic incident reporting and evidence management in Internet of Vehicles using machine learning and blockchain. *Engineering Applications of Artificial Intelligence,* p. 105630.
Politou, E. et al., 2020. Delegated content erasure in IPFS. *Future Generation Computer Systems,* pp. 956-964.
Politou, E. et al., 2022. Implementing Content Erasure in IPFS. *Privacy and Data Protection Challenges in the Distributed Era,* pp. 151-163.
Pongnumkul, S., Siripanpornchana, C. & Thajchayapong, S., 2017. Performance analysis of private blockchain platforms in varying workloads. *26th IEEE International Conference on Computer Communication and Networks (ICCCN) ,* pp. 1-6.
Prakken, H. & Sartor, G., 2015. Law and logic: A review from an argumentation perspective. *Artificial intelligence,* pp. 227, 214-245.
Ray, P. P., Dash, D., Salah, K. & Kumar, N., 2020. Blockchain for IoT-based healthcare: background, consensus, platforms, and use cases. *IEEE Systems Journal,* pp. 85-94.
Reuters, T., 2023. New report on ChatGPT & generative AI in law firms shows opportunities abound, even as concerns persist. *AI & Future Technologies.*
Ribeiro, M. T., Singh, S. & Guestrin, C., 2016. Model-agnostic interpretability of machine learning. *arXiv preprint,* Volume arXiv:1606.05386..
Rissland, E. & Ashley, K., 1987. A case-based system for trade secrets law. *Proceedings of the First International Conference on Artificial Intelligence and Law.*
Sachan, S., Almaghrabi, F., Yang, J. B. & Xu, D. L., 2021. Evidential reasoning for preprocessing uncertain categorical data for trustworthy decisions: An application on healthcare and finance. *Expert Systems with Applications,* pp. 185, 115597.
Sachan, S. et al., 2023. A Blockchain Framework in Compliance with Data Protection Law to Manage and Integrate Human Knowledge by Fuzzy Cognitive Maps: Small Business Loans. *In 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC),* pp. 1-4.
Sachan, S. & Muwanga, J., 2023. Integration of Explainable Deep Neural Network with Blockchain Technology: Medical Indemnity Insurance.
Sachan, S. et al., 2020. An explainable AI decision-support-system to automate loan underwriting. *Expert Systems with Applications,* p. 113100.
Sachan, S. et al., 2021. Augmented Intelligence for Transparent Decision Making in Insurance Claims. *In 31st European Conference on Operational Research.*
Seavey, W. A., 1942. Principles of torts. *Harv. L. Rev.,* Volume 56, p. 72.
Shacham, H. & Waters, B., 2013. Compact proofs of retrievability. *Journal of cryptology,* Volume 26(3), pp. 442-483.
Shah, M. A., Swaminathan, R. & Baker, M., 2008. Privacy-preserving audit and extraction of digital contents. *Cryptology ePrint Archive.*
Shinde, R., Patil, S., Kotecha, K. & Ruikar, K., 2021. Blockchain for securing ai applications and open innovations. *Journal of Open Innovation: Technology, Market, and Complexity,* p. 189.
Siddiqui, Z. A. & Haroon, M., 2023. Research on significant factors affecting adoption of blockchain technology for enterprise distributed applications based on integrated MCDM FCEM-MULTIMOORA-FG method. *Engineering Applications of Artificial Intelligence,* p. 105699.
Song, J. & Moon, Y., 2021. A Layer Image Auditing System Secured by Blockchain. *Procedia Manufacturing,* pp. 585-593.

Sunyaev, A., 2020. Distributed ledger technology. *Internet computing: Principles of distributed systems and emerging internet-based technologies,* pp. 265-299.
Sykes, A. O., 1988. The boundaries of vicarious liability: An economic analysis of the scope of employment rule and related legal doctrines. *Harvard Law Review,* Volume 101(3), pp. 563-609.
Taherdoost, H., 2023. Smart Contracts in Blockchain Technology: A Critical Review. *Information,* p. 117.
Torrey, D. B. & McIntyre, L. D., 2015. Recent Developments in Workers' Compensation and Employers' Liability Law. *Tort Trial & Ins. Prac. LJ ,* Volume 51, p. 749.
Union, E., 2016. Regulation (EU) 2016/679 (General Data Protection Regulation).
Ünsal, E., Örnek, H. K. & Tasdemir, Ş., 2023. A Review of Hashing Algorithms in Cryptocurrency. *In International Conference on Frontiers in Academic Research,* pp. 544-550.
Van Dis, E. A. et al., 2023. ChatGPT: five priorities for research. *Nature,* pp. 614(7947), pp.224-226.
Wöhrer, M. & Zdun, U., 2021. Architectural design decisions for blockchain-based applications. *IEEE International Conference on Blockchain and Cryptocurrency (ICBC),* pp. 1-5.
Wu, Y. L. X. L. X. S. J. C. P., 2016. A secure light-weight public auditing scheme in cloud computing with potentially malicious third party auditor. *IEICE TRANSACTIONS on Information and Systems,* Volume 99(10), pp. 2638-2642.
Xiao, Z. et al., 2023. Deep Contrastive Representation Learning with Self-Distillation. *IEEE Transactions on Emerging Topics in Computational Intelligence.*
Yang, J. B. & Xu, D. L., 2013. Evidential reasoning rule for evidence combination. *Artificial Intelligence,* pp. 205, pp.1-29.
Yang, W., Garg, S., Huang, Z. & Kang, B., 2021. A decision model for blockchain applicability into knowledge-based conversation system. *Knowledge-Based Systems,* p. 106791.
Zhang, W., Bai, Y. & Feng, J., 2022. Tiia: A blockchain-enabled threat intelligence integrity audit scheme for iiot. *Future Generation Computer Systems,* pp. 254-265.
Zhang, W., Bai, Y. & Feng, J., 2022. Tiia: A blockchain-enabled threat intelligence integrity audit scheme for iiot. *Future Generation Computer Systems,* pp. 254-265.
Zhu, H. et al., 2019. A secure and efficient data integrity verification scheme for cloud-IoT based on short signature. *IEEE Access,* Volume 7, pp. 90036-90044.
Zikratov, I. et al., 2017. Ensuring data integrity using blockchain technology. *20th Conference of Open Innovations Association (FRUCT),* pp. 534-539.