A Blockchain Solution for Decentralized Content Verification and its Application to Deepfake Detection and Fintech Credit Scoring

Luigi Coppolino, Giovanni Maria Cristiano, Salvatore D'Antonio, Jonah Giglio, Giovanni Mazzeo, and Luigi Romano[a]

[a]*University of Naples "Parthenope", Centro Direzionale, Isola C4, Naples, 80133, Italy*

Abstract

Ensuring the reliability and accuracy of information is a critical challenge in sectors like finance, media, and health. The reliance on centralized verification systems introduces risks of bias, manipulation, and limited transparency. To address these issues, we propose `VeriNet`, a decentralized framework for third-party content verification leveraging blockchain technology and the Ethereum Attestation Service. `VeriNet` integrates on-chain and off-chain attestations to ensure privacy, transparency, and accountability, supported by a Decentralized Data Warehouse and cryptographic Proof-of-SQL mechanisms. The framework includes *Contributors*, who submit content, and *Verifiers*, who assess its authenticity. We carried out two Proof-of-Concept implementations, namely in deepfake detection and fintech credit scoring, to demonstrate the efficiency of `VeriNet` and its adaptability to diverse domains. Moreover, we conducted an experimental evaluation focusing on various parameters, such as costs and execution time, to demonstrate framework feasibility, scalability, and potential to establish a trusted ecosystem for collaborative verification.

Keywords: Blockchain, Ethereum, Attestation, Deepfake Detection, Fintech, Credit Scoring, Artificial Intelligence

1. Introduction

In many sectors, it is increasingly important to verify content and resources through third parties to ensure objectivity and build public confidence. Information shared online — whether it is an image, a financial record, or a critical report — carries an inherent need for credibility (1). When this information is processed by one party alone, the risk of bias, manipulation, or errors becomes more significant (2). For example, in content verification, a single organization may lack the tools or expertise required to confirm whether an image or video has been digitally manipulated (3). Independent third-party experts can provide a fresh perspective, equipped with the necessary technology and insights to detect alterations that might otherwise go unnoticed. In cases involving sensitive or complex evaluations, like the credibility of a borrower's credit score, the process benefits from external validation. Third-party verifiers can assess the scoring methodologies used, ensuring that the criteria applied are fair, accurate, and free from conflicts of interest. Moreover, peer-based verification allows the introduction of specialized knowledge, which is often critical in technical or high-stakes fields. For instance, evaluating the authenticity of scientific research, intellectual property, or regulatory compliance may require expert oversight that a primary party alone might not be able to provide. With independent verifiers, claims can be rigorously tested against industry standards and best practices, building a more resilient foundation of trust. This is particularly relevant in areas like finance, media, and health, where accuracy and impartiality are not just ideal but essential.

In this paper, we propose `VeriNet` (i.e., *Verification Network*), a community-based framework for third-party verification that provides a structured way for experts, stakeholders, and other contributors to work together in confirming claims, creating a trusted ecosystem for verifying content and resources, and possibly earning rewards. By relying on third-party insights, organizations can produce verifiable information that meets a higher standard of trust, benefiting both providers and consumers in an information-driven world. Such a framework would not only enhance trust but also introduce accountability by making the verification process open and accessible. Our solution, `VeriNet`, uses the blockchain and an EVM-based attestation solution — namely *Ethereum Attestation Service (EAS)* — as foundational technologies to establish secure, transparent verification. Leveraging both *on-chain* and *off-chain* attestations from EAS, `VeriNet` adapts to varied verification needs. Off-chain attestations — whose metadata is anchored on-chain — allow us to maintain confidentiality where necessary, such as protecting the privacy of individual reviewers and their assessments. In contrast, on-chain attestations are used to record aggregated reviews, creating a tamper-resistant record that demonstrates the collective consensus. To foster active participation, `VeriNet` foresees a token-based incentive designed to reward *Verifiers* who contribute to the verification process.

`VeriNet` involves two key participant roles: *Contributors*, who submit content for verification, and *Verifiers*, responsible for assessing its authenticity or just suggesting different evaluations of *Contributors*' content. Both *Verifier* and *Contributors* generate off-chain attestations (whose timestamp is anchored on-chain) to ensure their privacy. These off-chain attestations are stored in a *Decentralized Data Warehouse (DDW)* to eliminate centralization and enable provable aggregation on the data using a cryptographic *Proof-of-SQL* process based on *Zero-*

Knowledge Proof (ZKP). Verifier evaluations are aggregated within the DDW and recorded as on-chain attestations for transparency and immutability. These attestations serve as verifiable certifications of content authenticity or quality.
We implemented a Proof-of-Concept of the framework for two different scenarios: *i*) deepfake detection through a community-based evaluation of image authenticity; *ii*) Fintech credit scoring based on peer evaluation of borrowers' credit. These scenarios were used to evaluate `VeriNet` in terms of cost and performance. Results demonstrate the feasibility of the proposed framework.
This paper provides a threefold contribution:

- It proposes a generic decentralized framework for community-based content verification, which can be applied to heterogeneous contexts.
- It provides a model that aims at rewarding verifiers for their contributions, encouraging active participation while enhancing transparency and credibility in the verification ecosystem.
- It demonstrates the applicability of the proposed framework to two different use cases, relying on a *de facto standard* technology for blockchain-based verification.

The remainder of this paper is organized as follows: Section 2 discusses the motivations behind the framework implementation and its potential applications. Section 3 presents the technologies utilized in this research work. Section 4 provides an overview of the related work in the field of content authenticity. Section 5 details `VeriNet` architecture and functionalities. Section 6 describes the Proof-of-Concept implementation, while in Section 7 results of the evaluation campaign are given. Section 8 addresses key operational and security issues. Finally, Section 9 summarizes the findings and concludes the paper.

2. Problem Domains

Ensuring the credibility and accuracy of information is a growing challenge in many domains, particularly as digital systems increasingly mediate interactions and decisions (4; 5; 6). To address these challenges effectively, we first examine the specific problem domains that motivate our research, as understanding the nature and scope of verification challenges is essential for designing appropriate technological solutions. In this section, we explore how `VeriNet` can address issues in two contexts: detecting fake images through multi-party analysis and refining credit scoring methodologies via collaborative evaluation. These examples highlight the need for a structured, decentralized verification process to foster trust, fairness, and accountability.

2.1. Deepfake Detection Challanges

The increasing prevalence of image manipulation techniques, such as deepfakes (7), poses serious challenges in domains where trust and accuracy are vital, particularly in social media and journalism. Advanced manipulation capabilities make it possible to alter images and videos in a way that can mislead audiences on a massive scale, impacting reputations, influencing public opinion, and undermining trust in reliable information sources (8). This is especially concerning in journalism, where manipulated visuals can distort facts, contribute to the spread of disinformation, and disrupt public confidence in media institutions (9; 10).
The rapid dissemination of falsified content on social networks increases these issues. On these platforms, altered images and videos can reach large audiences quickly. This leads to scenarios where users are exposed to misleading content without clear indicators of its authenticity, making it difficult to distinguish truth from false. The risks are high, as the effects of manipulated content go beyond mere disinformation. The reputations of individuals, companies, and institutions can be damaged, and the spread of false narratives can influence decision-making processes in various fields. Without effective strategies to counteract manipulation, the credibility of both social networks and journalistic sources remains at risk, creating a digital environment where reliable information is increasingly challenging to maintain.
A community-based verification framework, such as the one proposed in this paper, can be a key enabler in countering these challenges. By setting up a decentralized network of independent evaluators to collaboratively assess the authenticity of visual content, the framework could ensure a rigorous and transparent verification process.

2.2. Fintech Credit Scoring Issues

Credit assessments are a cornerstone of financial markets, enabling effective risk analysis and resource allocation. As of July 2024, the combined market capitalization of the largest Credit Ratings Agencies — i.e., S&P and Moody's — is almost a quarter of a trillion dollars. However, as the financial sector evolves with digital asset technologies and decentralized frameworks (11), traditional models face challenges related to scalability, transparency, and the mitigation of information asymmetry. The rapid adoption of tokenized Real-World Assets (RWAs) and the growth of private credit markets underline the demand for more accessible and reliable credit assessments (12). While tokenization offers technical efficiencies, liquidity in these markets depends on accurate risk evaluation and broad market participation. Existing frameworks in private credit rely heavily on opaque methodologies and limited access to high-quality information, resulting in market concentration and barriers to entry for smaller players.

The adoption of a credit scoring methodology shared by a community of experts could allow for the decentralisation of credit analysis, incorporating inputs from multiple stakeholders and fostering collaboration between issuers, analysts, and capital providers (13; 14). A multi-party credit evaluation may help address trust issues in credit assessment by aggregating expert opinions while preserving the confidentiality of underlying financial data. This supports a balanced approach that protects issuer privacy and enhances the credibility of credit ratings. Moreover, the collaborative model introduces traceability

and explainability into the credit assessment process, addressing key concerns for lenders and regulators alike. There are already initial efforts moving in this direction. For example, the Credora Inc.[1] company provides a privacy-preserving infrastructure that enables off-chain credit evaluations and on-chain attestations of credit scores. Their approach demonstrates how decentralised mechanisms can be adopted for credit analysis while maintaining confidentiality, transparency, and auditability.

3. Background

Building on the motivation and problem domains outlined in the previous section, we now present the core technological concepts that underpin our proposed solution. This section focuses on two essential components for our approach: the Ethereum Attestation Service and Space and Time, with its Proof of SQL protocol, which together provide the mechanisms necessary for decentralized, transparent, and verifiable computation.

3.1. Ethereum Attestation Service

Ethereum Attestation Service is a protocol that enables the creation, management, and verification of attestations on Ethereum and EVM-compatible blockchains (15). An attestation, in the EAS context, represents a digitally signed statement made by an issuer about any subject, entity, or claim. EAS provides a standardized, permissionless framework for creating verifiable proofs about virtually anything—from identity credentials and ownership claims to skills, achievements, and access rights (16). The EAS architecture, illustrated in Figure 1, consists of three main components that work together to provide a flexible attestation system: the participants, the Schema Registry Contract, and the Attestation Contract[2].

The Schema Registry Contract serves as a decentralized repository for attestation templates. Before creating attestations, issuers must register a schema that defines the data structure and fields required for the attestation, the data types for each field (such as address, uint256, string, or bool), whether attestations following this schema can be revoked, and an optional resolver contract address for custom validation logic. Each registered schema receives a unique identifier (UID) that attestations reference when being created.

The Attestation Contract represents the core component responsible for creating, storing, and managing attestations. When an issuer creates an attestation, they must specify the schema UID to follow, the recipient's Ethereum address (which is optional), the attestation data encoded according to the schema, an optional expiration time, and potentially a reference to another attestation for creating attestation chains. This contract ensures that all attestations adhere to their specified schemas and maintain the integrity of the attestation system. A powerful component is the Resolver Contract, a custom smart contract

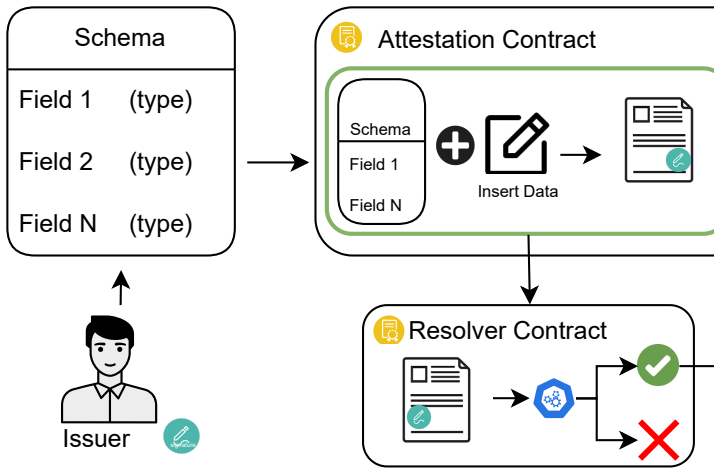

Figure 1: Ethereum Attestation Service architecture showing the interaction between smart contracts, attestation types, and participants

that can be associated with a schema to enforce additional validation rules or business logic. When an attestation is created or revoked, the resolver contract is automatically invoked to validate attestation data against custom criteria. If the resolver contract's validation fails, the entire attestation transaction is reverted, ensuring only valid attestations are recorded. EAS supports multiple attestation paradigms to accommodate different privacy and accessibility requirements[3]. On-chain attestations are stored directly on the Ethereum blockchain, providing immutability and permanent availability, public verifiability without requiring external dependencies, automatic integration with smart contracts and DApps, though at higher gas costs due to on-chain storage. For privacy-sensitive use cases, off-chain attestations can be created and stored externally while maintaining cryptographic verifiability. This approach ensures that data remains private and is shared selectively, incurs significantly lower costs as there are no gas fees for storage, and can be anchored on-chain via timestamps for temporal proof, though it requires external storage and availability considerations. A particularly valuable feature for various use cases is Delegated Attestations, which enables separation of signing and fee payment responsibilities. In this model, the attestor signs the attestation with their private key while a different entity, the delegator, submits the transaction and pays gas fees. This mechanism is particularly useful for subsidizing user attestations or bulk processing while maintaining cryptographic proof of the original attestor's identity. Additionally, Referenced Attestations allow attestations to reference other attestations, creating attestation chains for versioning and updates, hierarchical relationships between claims, audit trails showing the evolution of attestations, and complex attestation graphs for multi-party scenarios.

The typical workflow for creating and using attestations begins with schema registration, where an issuer designs and registers a schema that defines the structure for their attestation type.

[1] `https://www.credora.network/`
[2] https://docs.attest.org/docs/core–concepts/how-eas-works
[3] https://docs.attest.org/docs/core–concepts/attestations

This is a one-time operation per attestation type. Following this, the attestation creation process involves encoding data according to the registered schema, signing the attestation either directly or via delegation, submitting the transaction to the attestation contract, and optionally invoking the resolver contract validation. Once created, on-chain attestations are permanently stored on Ethereum, while off-chain attestations are stored externally with optional on-chain timestamps, and each attestation receives a UID. The verification process is accessible to anyone. It involves checking the cryptographic signature of the issuer, validating the attestation against the registered schema structure, confirming the attestation has not been revoked, and verifying it has not expired if an expiration was set. When attestations are no longer valid, issuers can revoke them, with the revocation permanently recorded on-chain, ensuring the system always reflects the current state of claims.

3.2. Decentralized Data Warehouse and Proof-of-SQL

A Decentralized Data Warehouse represents a paradigm shift from traditional centralized data storage systems. Unlike conventional data warehouses that rely on a single controlling entity, a DDW distributes data storage and processing across a network composed of various nodes. The fundamental principle is to remove centralized control while preserving the functionality and performance characteristics that make data warehouses valuable for analytical workloads.

A challenge emerged in decentralized data warehousing regarding the verification of query execution correctness and data integrity. This necessitates the introduction of *Proof of SQL* (17), a zero-knowledge cryptographic protocol that provides mathematical guarantees about the correctness of database query execution[4]. The protocol builds upon several fundamental cryptographic and mathematical concepts that work together to enable verifiable computation over databases. At its theoretical foundation, *Proof of SQL* relies on polynomial commitment schemes[5]. The key insight is that database tables can be represented as polynomials, where each row and column corresponds to specific polynomial evaluations. This mathematical transformation allows the system to leverage powerful algebraic properties: any change to the underlying data would result in a completely different polynomial, making tampering mathematically detectable. The protocol employs multilinear extensions, which provide a way to represent discrete database tables as continuous mathematical functions. This representation enables the system to perform computations on the data while maintaining cryptographic properties. When a SQL query is executed, it corresponds to specific polynomial operations that can be verified without revealing the underlying data values. Zero-knowledge proof systems, particularly those based on interactive protocols, form another crucial foundation. The system uses these to generate proofs that demonstrate correct query execution without revealing any information beyond the correctness of the result (18). The prover (the node executing the query) can convince a verifier that the computation was performed correctly, while the verifier learns nothing about the actual data or intermediate computation steps. The protocol incorporates sum-check protocols, which allow verification of polynomial computations through a series of challenges and responses. When a SQL query involves aggregations, joins, or other operations, these can be expressed as polynomial relationships that the sum-check protocol can verify efficiently. This enables the system to handle complex database operations while maintaining cryptographic guarantees. Commitment schemes provide the mechanism for creating tamper-evident "fingerprints" of database columns. They bind the prover to specific data values without revealing those values, and any attempt to use different data would require breaking fundamental cryptographic assumptions. The commitments are updatable, meaning new data can be added to tables without requiring the complete recomputation of all commitments. The verification process relies on constraint systems that encode the correctness of SQL operations as mathematical relationships. Each type of database operation (selection, projection, join, aggregation) corresponds to specific polynomial constraints that must be satisfied for the computation to be considered correct.

3.3. Requirements Elicitation

Traditional content verification typically relies on centralized authorities or single-party assessments. For instance, in journalism, fact-checking is often performed by individual organizations with limited transparency (19). In credit scoring, assessments depend on opaque methodologies from established agencies (20; 21). These approaches suffer from potential bias, lack of transparency, and single points of failure.

To address these limitations, the following requirements must be satisfied:

R1. *Decentralized Verification*: The system must enable multiple independent verifiers to collaboratively assess content authenticity without relying on a single centralized authority, ensuring no single point of failure or bias.

R2. *Tamper-Proof Data Storage*: All attestations and verification data must be stored in a tamper-resistant manner ensuring data integrity throughout the verification process.

R3. *Privacy-Preserving Evaluations*: Individual verifier evaluations must remain confidential while still enabling transparent aggregation of results, protecting verifier privacy and preventing influence between assessments.

R4. *Verifiable Aggregation*: The aggregation of verifier assessments must be cryptographically provable, allowing anyone to verify the correctness of the consensus without accessing individual evaluations.

R5. *Incentive Mechanism*: The framework must provide a feature for rewarding verifiers, encouraging active participation and discouraging malicious behavior.

R6. *Traceability and Accountability*: All verification activities must be traceable through attestation relationships and

[4]https://docs.spaceandtime.io/docs/overview-1

[5]https://docs.spaceandtime.io/docs/trustless-database-for-the-evm

unique identifiers, enabling audit trails while maintaining the ability to revoke and cascade invalidation when necessary.

R7. *Cross-Domain Adaptability*: The system must support different types of content verification (images, video, audio, financial data) through flexible schema definitions while maintaining consistent verification workflows across domains.

R8. *Scalability and Performance*: The framework must handle varying numbers of participants efficiently, with costs that scale based on system usage and complexity.

Category	Query
Decentralized content verification	`TITLE-ABS-KEY(("content verification" OR "information authenticity" OR "data verification") AND blockchain AND (decentralized OR "distributed systems"))`
Community-based verification	`TITLE-ABS-KEY(("community-based verification" OR "crowdsourced verification" OR "peer verification") AND blockchain AND (trust OR reputation OR validation))`
Blockchain attestations	`TITLE-ABS-KEY(("blockchain attestation" OR "blockchain-based attestation" OR "smart contracts for attestations" OR "Ethereum Attestation Service" OR "Proof-of-SQL"))`
Decentralized image authenticity	`TITLE-ABS-KEY(("image authenticity" OR "video authenticity" OR deepfake OR "media verification") AND blockchain AND (decentralized OR "distributed ledger"))`
Community-based credit scoring	`TITLE-ABS-KEY(("credit scoring" OR "reputation scoring" OR "financial trustworthiness") AND blockchain AND ("community-based" OR "peer-based" OR decentralized))`

Table 1: Queries grouped by research category

4. Related Work

To better understand the state of the art in decentralized content verification, we conducted an in-depth search using specific queries, summarized in Table 1. These queries allowed us to gather relevant research on various aspects of the field, from blockchain-based attestations to community-driven and financial applications. The increasing need for trustworthy content verification in decentralized environments has motivated extensive research into blockchain-based solutions. Our literature review used Scopus and IEEE Xplore databases, showing a total of 154 papers. After applying filtering criteria including duplicate removal, temporal constraints (publications within the last 10 years), language requirements (English only), and relevance screening, we identified the most pertinent works for our analysis. This systematic approach was supplemented with backward and forward citation analysis to capture additional relevant studies that informed our understanding of existing solutions and research gaps. In the following analysis, we evaluate each work against the eight requirements defined in Section 3.3
Early work by Härer and Fill (22) demonstrated how blockchain attestations can bind digital artifacts to user identities, establishing foundational concepts for model-based attestations and long-term verifiability. These authors later expanded their idea in (23), proposing a more comprehensive system that combines on-chain attestations with off-chain storage protocols like IPFS, addressing both data integrity and availability challenges. Building on these attestation concepts, He et al. (24) present a novel approach that leverages smart contracts and game theory to incentivize honest behavior in database systems, effectively eliminating the need for trusted intermediaries. The challenge of maintaining accountability in collaborative environments is addressed by Zhou et al. (25) who introduce mechanisms for confidential auditing in distributed workflows while preserving data privacy.
Recognizing that automated systems alone were insufficient, researchers turned to blockchain-enabled collective intelligence approaches. Paul et al. (26) proposed a system that leverages distributed human validation through a decentralized network, where validators are assigned weights based on their reputation. The synthesis of human and artificial intelligence reaches maturity in the work of Butincu et al. (27), which presents a comprehensive platform balancing automated AI analysis with human judgment. These collective intelligence approaches benefit significantly from standardized attestation infrastructure, as exemplified by the Ethereum Attestation Service described in (16), which provides unified schemas and interfaces for creating and managing attestations.
The emergence of deepfake technology has created urgent demands for media authentication systems. Early efforts by Hasan and Salah (28) demonstrate how video hashes stored on Ethereum can enable basic authenticity verification. However, the sophistication of deepfake technology necessitated more advanced approaches, leading to systems like those proposed by Costales et al. (29) and the BAAI framework (30), which integrate convolutional neural networks and ensemble methods with blockchain verification. The HyperSwin framework (31) represents the current state of the art, combining Swin Transformers with Hyperledger Fabric to enable collaborative detection among trusted parties through private blockchain networks.
These verification principles extend naturally to financial applications. Zhang et al. (32) demonstrate how consortium blockchains can facilitate secure credit information sharing while addressing privacy concerns through permission management. The behavioral aspects of financial decision-making are explored by Hassija et al. (33), who integrate prospect theory with blockchain to create more nuanced credit assessment systems. The latest advances by Jovanovic et al. (34) address the critical need for model explainability and verification in automated credit decisions, combining federated learning with blockchain to ensure both privacy and transparency. As shown in Table 2, while existing approaches address various subsets of the identified requirements, no single solution comprehensively covers all eight requirements simultaneously. Most works focus on fundamental aspects like R1 (Decentralized Verification) and R2 (Tamper-Proof Data Storage), which are commonly implemented across the analyzed solutions. However, critical requirements such as R5 (Incentive Mechanism) and R4 (Verifiable Aggregation) are addressed by only a limited number of existing solutions. This analysis reveals significant gaps in current approaches, particularly regarding incentivization mechanisms and cryptographically verifiable aggregation processes.

With respect to the existing literature, our proposed frame-

Paper	R1	R2	R3	R4	R5	R6	R7	R8
Härer & Fill (22)	✓	✓	-	-	-	✓	✓	-
Härer & Fill (23)	✓	✓	-	-	-	✓	✓	✓
He et al. (24)	✓	✓	-	-	✓	-	-	✓
Zhou et al. (25)	✓	✓	✓	-	-	✓	-	-
Paul et al. (26)	✓	-	-	-	-	-	✓	-
Butincu et al. (27)	✓	-	-	-	✓	-	✓	-
Hasan & Salah (28)	-	✓	-	-	-	✓	✓	-
Costales et al. (29)	-	✓	-	-	-	-	✓	-
Priya et al. (30)	✓	✓	-	-	-	-	✓	-
Bindra et al. (31)	✓	✓	✓	-	-	✓	✓	✓
Zhang et al. (32)	✓	✓	✓	-	-	✓	-	✓
Hassija et al. (33)	✓	✓	-	-	✓	-	-	✓
Jovanovic et al. (34)	✓	✓	✓	✓	-	✓	-	✓
`VeriNet` **(Our work)**	✓	✓	✓	✓	✓	✓	✓	✓

Table 2: Requirements Addressed by State-of-the-Art Solutions

work presents a more versatile solution that can be applied across a range of contexts, extending beyond the verification of specific content types like videos or documents. A difference is the design of the framework, which enables a more efficient and flexible verification process by including methods that are not limited to human-based validation. Furthermore, our framework addresses a limitation that is present in various prior works, such as the absence of a mechanism to incentivize the participants. By introducing a structured reward system, we ensure active and sustained participation in the verification process. Moreover, we use a *de facto standard* technology for blockchain-based verification, i.e., the Ethereum Attestation Service.

5. `VeriNet`

In this section, we present the `VeriNet` framework including a description of its high-level features and a more detailed analysis of a typical workflow.

5.1. Overview

`VeriNet` is designed to create a fully decentralized, reliable, and transparent system for verifying and certifying content.
We foresee the existence of two types of participant entities: *Contributors* and *Verifiers*. *Contributors* are content creators or data owners, who initiate the verification process. *Verifiers* are third-party entities responsible for evaluating the content's authenticity. They are incentivized through a token-based reward system, which ensures ongoing participation and sustainability within the framework. Figure 2 provides a high-level overview of `VeriNet`.
A key component of the framework is the *Ethereum Attestation Service*. *Contributors* leverage the EAS service to create an off-chain attestation — whose timestamp is anchored on-chain — that includes the content they wish to verify. The contributor attestation follows a pre-defined schema, which is publicly available in the EAS Schema repository. Once the off-chain attestation is created, it is sent to a *Decentralized Data Warehouse* for storage ①. The reason behind the adoption of a decentralized data warehouse lies in the need to eliminate any centralization point. Furthermore, we foresee a data warehouse that enables provable computations performed on the stored data.
Verifiers read the attestation submitted by the *Contributors* ② and, after performing necessary calculations, assign a score or any other assessment metric of the content's authenticity ③. These assessment metrics can include confidence scores (e.g., a percentage indicating the likelihood of content authenticity), numerical ratings (e.g., credit scores or risk assessments), binary classifications (e.g., authentic/fake, approved/rejected), or categorical evaluations (e.g., low/medium/high risk levels). The specific metric type depends on the verification domain and the nature of the content being evaluated. This evaluation is also anchored inside an off-chain attestation and stored in the Decentralized Data Warehouse ④. These verifiers' attestations are then aggregated using SQL queries whose correctness is ensured by *Proof-of-SQL* verification process ⑤, which cryptographically ensures that SQL queries were computed correctly against unmodified data. The *Verifier* evaluations, along with the proof commitment, are attached to an on-chain attestation and stored immutably and transparently on the blockchain ⑥. If a new *Verifier* comes in, the aggregated attestation is computed again. *Contributors* can consume the on-chain attestations via the EAS service and use them as a certification of authenticity and quality ⑦.

5.2. Workflow

5.2.1. Participant Definitions

Let the *Contributors* be represented by the set C, which consists of entities or individuals who submit content for validation, aiming to establish its authenticity through an impartial verification process. Each *Contributor* submits a content piece from the set of all possible content types, $\mathcal{X}$.
The content can vary, including types such as images, audio, video, or text, which require impartial validation for authenticity. The type of contributor may vary based on the context. They can range from general users to professionals with varied expertise, depending on the content that needs validation.
Verifiers, represented by the set $\mathcal{V}$, are responsible for evaluating and confirming the validity of the content using specific standards.
Verifiers are split into two subsets: *Manual Verifiers* ($\mathcal{V}_M$) and *Automated Verifiers* ($\mathcal{V}_A$). This division allows for both human-led and algorithmic verification: $\mathcal{V} = \mathcal{V}_M \cup \mathcal{V}_A$
Manual Verifiers are typically experts or trusted community members who conduct human evaluations. *Automated Verifiers* use algorithms, such as neural networks and other AI models, to execute verification tasks efficiently.
In some cases, these two verifier types collaborate for improved accuracy. For instance, a *Manual Verifier* may review the results generated by an *Automated Verifier*, intervening if anomalies or ambiguous outcomes are detected.
Every verifier must deposit a certain number of tokens as a guarantee of their commitment. This process, known as *staking*, ensures that only serious and trustworthy participants are involved. Similar to Proof-of-Stake (PoS) systems (e.g. Ethereum), staking serves as a warranty to discourage dishonest behavior. If a verifier submits false or misleading, they could be penalized through *slashing*, meaning they lose a part of their staked tokens. This mechanism is fully described in Section 5.3.

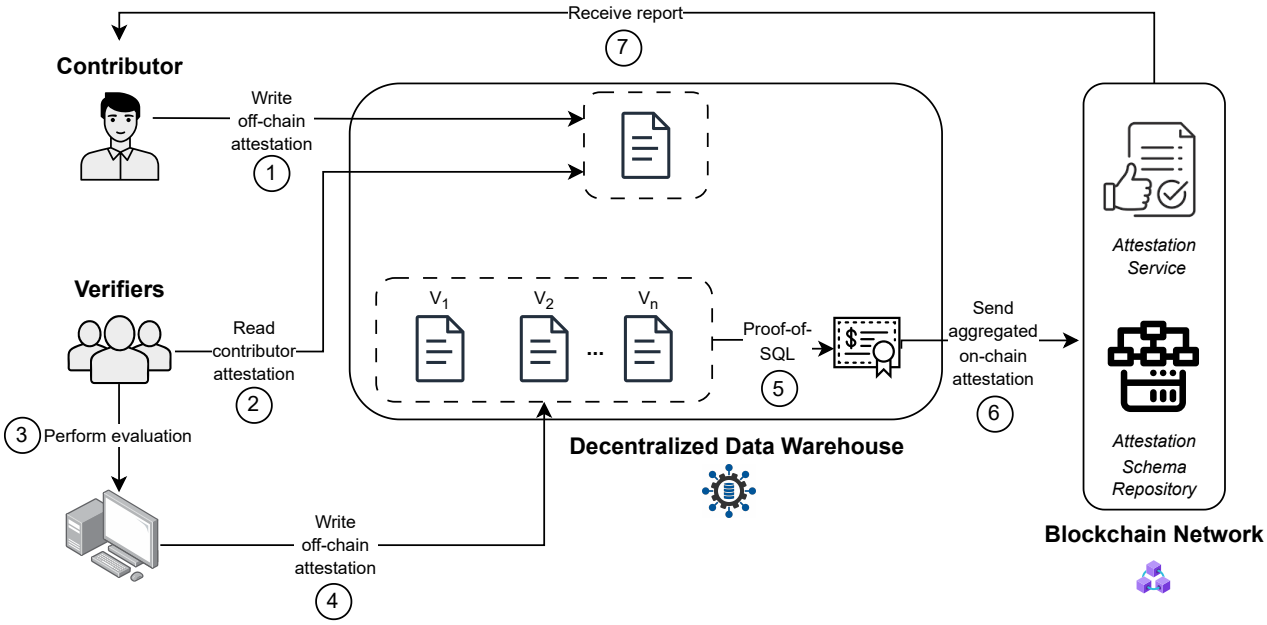

Figure 2: High-level architecture of `VeriNet`

5.2.2. Contributor Off-Chain Attestations

As depicted in Figure 3, the verification process starts when a *Contributor* submits content to verify its authenticity. The *Contributor* generates an off-chain attestation O_i^{off}, which can store various parameters. Among these parameters, the attestation includes the content that the *Contributor* has to verify.

For the off-chain attestation process, `VeriNet` employs the *EAS Contributor Schema*, linked to a particular resolver address. The resolver address is integral to the attestation framework, as it verifies the registration status of entities within the network. Only participants who are formally registered are allowed to take part in the validation process. Each participant in the network is assigned a unique address.

The resolver address verifies if a given address corresponds to an authorized participant within `VeriNet`. For each *Contributor*, the resolver address also functions as the recipient address at the end of the attestation process. This address holds a historical record of all content submissions by each *Contributor*, providing traceability within the `VeriNet` framework.

5.2.3. Off-Chain Attestations Storage

Off-chain attestations are stored in a DDW. The DDW serves a dual purpose: it provides decentralized, secure data storage and supports computational operations through queries. Moreover, it generates verifiable proofs that confirm the integrity of the operations performed. These proofs act as immutable records, enabling users to verify both the accuracy of the data and the operations conducted on it. This approach is essential for ensuring transparency and trust in the data handling process. The DDW's tamper-resistant and secure structure ensures that stored data remains unchanged over time. We foresee a DDW, with support for verifiable query execution through the *Proof-of-SQL* protocol.

The DDW contain both the single off-chain attestation and a cryptographic proof π that certifies the query was executed over unaltered data. As discussed, this proof-of-SQL π is generated using a zero-knowledge primitive and can be verified independently by any participant.

The DDW can be formally described by the tuple:

$$W = (D, H, Q, P)$$

where D is the dataset, H is a hash function providing content-addressability, Q is the set of supported SQL-like queries, and P is the proof-generation mechanism executed on the aggregation queries (discussed in the next subsection).

5.2.4. Aggregated On-Chain Attestations

Once the attestations are submitted, they are stored in the DDW. This storage not only ensures the integrity and traceability of the data but also lays the foundation for the next step: aggregation.

To provide the *Contributor* with a unique result, the DDW aggregates the N attestations using predefined rules. A classical method of aggregation is to calculate the arithmetic mean of the scores S_m submitted by the *Verifiers*. However, to incorporate the historian of the *Verifiers*, the formula is adjusted to include the historical performance score H_m. The aggregated score is computed as a weighted average:

$$S_{\text{avg}} = \frac{\sum_{m=1}^{N} H_m S_m}{\sum_{m=1}^{N} H_m}.$$

In this formula, each score S_m is weighted by the corresponding historical score H_m, which reflects the *Verifier*'s past alignment

with the consensus. The methodology for calculating H_m will be discussed in detail in Section 5.3.

Let us consider a concrete example: a Contributor submits a content and an off-chain attestation that includes fields like `contentHash`, `submissionTime`. This attestation is stored as an entry d_i in the DDW dataset D. Later, three Verifiers evaluate the content and submit their own attestations, each including an `score`. These are stored as entries d_{v1}, d_{v2}, d_{v3}. A sample aggregation query could be the following:

```
SELECT SUM(score * weight) / SUM(weight) AS weighted_avg
FROM VerifierAttestations
WHERE contentHash = '0xabc...'
```

This query retrieves all rows in the `VerifierAttestations` table where the field `contentHash` equals `0xabc...`, and computes the weighted average of the field `score`. The Proof-of-SQL system guarantees that both the filtering condition and the aggregation function have been executed over the original, unaltered dataset. Internally, the proof is constructed as follows:

1. Query Rewriting and Compilation. The SQL query is translated into an arithmetic circuit or intermediate representation.

2. Data Commitment. The DDW maintains a Merkle tree over the table. Each row has a commitment (hash) and is linked to the table's root hash r_D. This ensures immutability of the dataset.

3. Execution Trace. The ZKP system generates a trace of the SQL execution, proving that:

   - the selected rows all match the filter

     the sum and count were computed correctly over these rows

   - the data used in the computation is included in the committed dataset D

4. Proof Generation. Using the execution trace and the committed state, the DDW generates a succinct, non-interactive proof π satisfying:

   $$\text{Verify}(\pi, q, r_D) = \text{true}$$

   meaning that query q was executed correctly over D whose root hash is r_D.

Once the aggregation is complete, the consensus is reached, and the proof is generated, the final result is then recorded as an on-chain attestation, which ensures the immutability of the outcome and enables public verification. To ensure transparency, the on-chain attestation also contains the proof commitment — i.e., the cryptographic evidence — that certifies the correct aggregation process.

Upon receiving an on-chain aggregated attestation (which includes the aggregated score and the hash of π), the Contributor can independently query the DDW using the same query and request a fresh proof π'. The Contributor verifies that $H(\pi') = H(\pi)$ matches the hash committed in the on-chain attestation. This can be achieved by running the proof verifier locally to ensure that π' is a valid proof of query q over the dataset D.

In this way, Contributors do not need to trust the entity who executed the query. Instead, they rely on the validity of the cryptographic proof. In addition, since the raw data and proofs are verifiable, external auditors or observers can validate the aggregated outcome.

5.2.5. Relationships Between Attestations

Attestations are structured through specific relationships involving *Contributors*, *Verifiers*, and *Aggregated Attestations*. These relationships leverage the concepts of *referenced attestations* and the *recipient* field as defined in Section 3.

Initially, a *Contributor* creates an off-chain attestation without any recipient or reference. The payload contains the content to be verified, and the attestation does not reference any prior attestations. The recipient is unspecified for initial *Contributor* attestations.

If the *Contributor* later decides to revise their attestation content, they create a new attestation referencing the unique identifier of their previous attestation.

Verifiers create off-chain attestations based on their evaluation of a contributor's attestation. The verifier's attestation references the unique identifier of the contributor's attestation and uses the contributor's wallet address as the recipient. The payload contains the verifier's assessment or score. This structure ensures that verifier attestations are directly tied to specific contributor attestations, providing traceability and accountability.

Aggregated attestations are created on-chain to summarize the results of verifier attestations. They reference a set of off-chain verifier attestations and use the contributor's wallet address as the recipient. The aggregated payload contains results such as a mean score or consensus result. This composite reference to multiple verifier attestations ensures that the aggregated attestation is tied to all the underlying evaluations.

5.2.6. Revocation and Cascading Effects

Revocation policies in the `VeriNet` framework ensure the integrity of attestations by handling changes or errors in data. *Contributors* can revoke their attestations if their data is updated or incorrect. *Verifiers*' attestations are invalidated if the *Contributor*'s attestation is revoked, ensuring cascading invalidation of dependent data. *Verifiers* can also revoke their attestations if they identify errors or wish to update their inputs. The structured relationships between attestations enable efficient management of revocations and their cascading effects.

The revocation does not remove an attestation from the blockchain—since data on-chain is immutable—but instead marks the attestation as revoked by updating its status in the corresponding smart contract that stores attestations. This allows clients and applications to identify and disregard revoked attestations without altering historical data.

We customized the revocation process by including a cascading revocation. If a *Contributor* revokes an attestation a_c, all

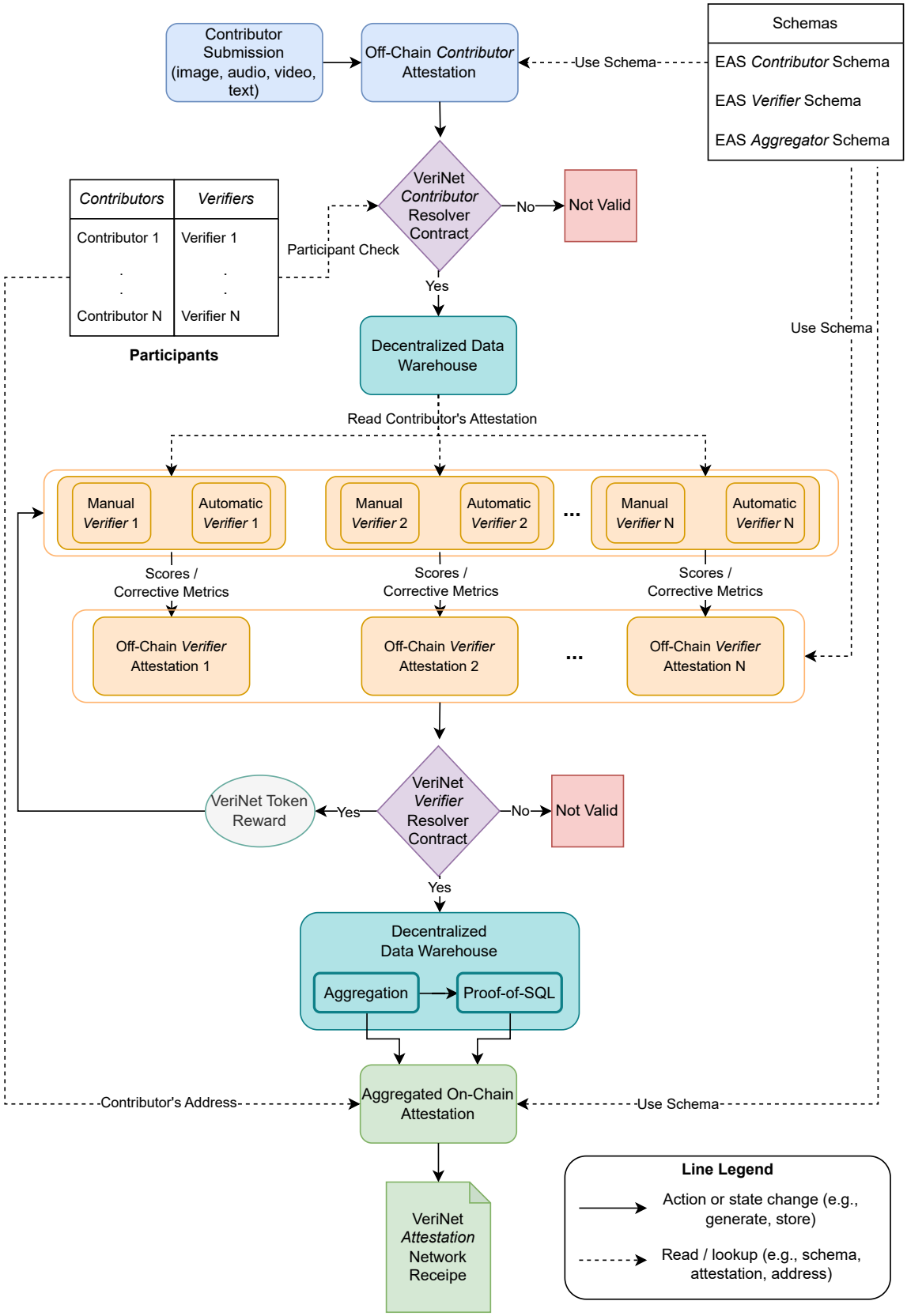

Figure 3: The VeriNet Workflow

verifier attestations a_v that reference a_c are automatically considered invalid. Similarly, if a *Verifier* revokes an attestation a_v, all aggregated attestations a_a that reference a_v are invalidated. These cascading effects ensure that the integrity of the framework is preserved by preventing revoked or invalid data from propagating through the system.

5.3. Rewarding Mechanism

To ensure fair and effective verification, `VeriNet` rewards *Verifiers* using a hybrid approach based on both historical accuracy and alignment with consensus in each verification session. This mechanism encourages accurate assessments while penalizing inconsistent or extreme evaluations.

5.3.1. Measuring Deviation from Consensus

The deviation d_m for each *Verifier* is computed as:

$$d_m = |S_m - S_{\text{avg}}|$$

This deviation determines how well a *Verifier* aligns with the consensus.

5.3.2. Historical Performance Score

Each *Verifier* has a historical performance score H_m, updated dynamically based on past accuracy. It is computed as an exponentially weighted moving average (EWMA) (35) of past alignment scores:

$$H_m^{(t)} = \lambda H_m^{(t-1)} + (1 - \lambda) f(d_m)$$

where λ is a decay factor (e.g., 0.8) and $f(d_m)$ is a function that decreases as deviation increases:

$$f(d_m) = e^{-\alpha d_m}$$

where α controls the penalty for deviation.

5.3.3. Computing the Consensus Score

Each *Verifier* v_m provides a score S_m for a given verification task. The consensus score S_{avg} is computed as a weighted average, where weights are given by the historical performance scores H_m:

$$S_{\text{avg}} = \frac{\sum_{m=1}^{N} H_m S_m}{\sum_{m=1}^{N} H_m}.$$

5.3.4. Staking and Reward Calculation

Each *Verifier* is required to stake an amount of tokens as collateral for their participation. This stake acts as a security deposit, ensuring commitment to accurate verification. The total reward r_m is computed as:

$$r_m = R \cdot L_m \cdot f(d_m) \cdot P_m$$

where:

- R is the base reward.
- L_m is the historical performance multiplier, defined as:

$$L_m = \begin{cases} L_{\text{base}} & \text{if } H_m < H_{\text{threshold, base}} \\ L_{\text{intermediate}} & \text{if } H_{\text{threshold, base}} \leq H_m < H_{\text{threshold, advanced}} \\ L_{\text{advanced}} & \text{if } H_m \geq H_{\text{threshold, advanced}} \end{cases}$$

- $f(d_m)$ scales the reward based on how close v_m is to the consensus.
- P_m is a participation consistency factor, rewarding users who participate regularly:

$$P_m = \frac{\text{tasks completed in last 30 days}}{\text{max tasks by any verifier in 30 days}}$$

A higher stake may be required for verifiers with a low historical score, increasing their financial commitment to prevent manipulative behavior. Staking also enables slashing, enforcing accountability for inaccurate assessments.

5.3.5. Slashing Mechanism

To discourage incorrect assessments, a slashing mechanism is applied. If a *Verifier*'s deviation d_m exceeds a predefined threshold $d_{\max}$, they receive no reward and a portion of their staked amount is slashed:

$$r_m = 0, \quad S_m = \gamma S_m^{(t-1)}$$

where S_m is the staked amount, and γ (e.g., 0.9) reduces the stake, penalizing poor performance. Additionally, their historical score is penalized:

$$H_m^{(t)} = \beta H_m^{(t-1)}$$

where β (e.g., 0.9) further decreases their historical credibility, discouraging misaligned evaluations.

6. Proof-of-Concept

This section presents the practical application of the proposed framework for the two case studies presented in sections 2.1-2.2. These are representative because they show the adaptability of the framework to different domains and computational requirements.

The implementation of `VeriNet` uses the Ethereum Sepolia blockchain and leverages the DDW offered by *Space&Time*, which provides a built-in open-source solution for generating *Proof-of-SQL* commitments.

The implementation details described in this section are fully documented. The complete VeriNet framework, including smart contracts, is publicly available on GitHub[6]. The repository includes deployment scripts, configuration files, and comprehensive documentation for reproducing the experimental results.

[6] `https://github.com/giammycri/VeriNet`

Schema	Field Name	Type
Content Creator	contentHash	bytes32
	contentType	string
	metadataURL	string
	submissionTime	uint256
	deviceType	string
Analyzer	algorithmName	string
	algorithmDetailsURL	string
	weightsFileURL	string
	accuracyScore	uint256
Aggregated Review	aggregatedScore	uint256
	proofCommitment	bytes32
	numberVerifiers	uint256
	finalVerificationResult	string
	aggregationMetricType	string

Table 3: Attestation Schemas for the Deepfake Detection Case Study

6.1. Case Study 1: Deepfake Detection

This case study illustrates how `VeriNet` addresses challenges in content authenticity, particularly the deepfake problem. The framework involves two main participant roles: *Content Creators* and *Analyzers*. The schema for these roles, along with the specific fields used, is detailed in Table 3.

Content Creators are responsible for submitting multimedia content, such as videos or images, for deepfake analysis. Their role corresponds to the *Contributor* role, and their primary responsibility is to provide detailed metadata about the submitted content. This process is facilitated by the Content Creator Schema, which includes key fields such as *contentHash*, a cryptographic hash that ensures the integrity and uniqueness of the submitted content, and *contentType*, which specifies the type of media (e.g., image, video). Additional metadata is saved through the *metadataURL* field, which links to contextual information, and the *submissionTime* field, which records the precise timestamp of the submission to enhance traceability. Moreover, the *deviceType* field specifies the device used to create the content, further supporting the verification process by providing information about its origin. All this metadata is securely stored in an off-chain attestation, ensuring reliable and tamper-proof access for subsequent analysis. By employing this schema, the framework ensures that submissions are well-documented and standardized.

Analyzers — i.e., *Verifiers* — are automated systems that use advanced AI models and machine learning algorithms to detect potential manipulations indicative of deepfakes. These systems analyze the content for patterns, inconsistencies, and subtle markers that may escape traditional detection methods. Their outcomes are attached to an off-chain attestation following the *Analyzer Schema*. This schema includes the *algorithmName* field, identifying the specific algorithm or model used, and *algorithmDetailsURL*, which links to documentation about the detection methodology. To ensure reproducibility and transparency, the *weightsFileURL* field references the trained model's parameter files. The results of the analysis are quantified through the *accuracyScore* field, which provides a confidence measure for the automated findings. Analyzers generate a detailed report containing these findings, which is then passed to the next stage for aggregation. The Analyzer attestation will also insert the wallet address of the content creator in the *Recipient* field of the EAS off-chain attestation. In this way, there is a clear track of attestation targets in the network.

The scores and findings are aggregated using the *Aggregated Review Schema*, which consolidates results from multiple automated analyzers. This schema includes the *aggregatedScore* field to reflect the combined confidence levels from all *Verifiers* and the *proofCommitment* field to secure the aggregated findings cryptographically. The *numberVerifiers* field tracks the number of analyzers contributing to the aggregated result, while the *finalVerificationResult* field provides a clear and definitive conclusion regarding the authenticity of the submitted content. Additionally, the *aggregationMetricType* field specifies the metric used to combine the scores, ensuring transparency and clarity in the aggregation process. These aggregated results are compiled into an on-chain attestation, providing a definitive and trustworthy evaluation regarding the authenticity of the submitted content.

Schema	Field Name	Type
Borrower	financialStart	string
	financialEnd	string
	transDescr	string
	activeLoans	string
Scorer	methodology	bytes32
	borrowerAddr	address
	borrowerUid	string
	benchmarking	string
	score	uint256
	borrowCapacity	uint256
	probabilityDefault	uint256
	KYB	bool
	financial	bool
	notes	string
Reviewer	anchorAdjustment	bytes32
	modifiers	string
	totalAdjustment	string
	probabilityDefaultAdj	uint256
	keyRisks	string
	notes	string
Aggregated Review	aggAtts	string
	probabilityDefaultAdj	uint256
	notes	string
	proofCommitment	string

Table 4: Attestation Schemas for the Fintech Credit Scoring Case Study

6.2. Case Study 2: Fintech Credit Scoring

The `VeriNet` framework can enable transparent and reliable credit assessments by leveraging the attestation schemas detailed in Table 4. These schemas define the data fields associated with each participant role, i.e., *Borrowers*, *Scorers*, *Reviewers*. *Borrowers* — i.e., the equivalent of *Curators* — initiate the process by submitting their financial and operational details, which are encapsulated in the *Borrower* schema. This schema includes fields such as *financialStart* and *financialEnd* to denote the timeframe of their financial records, *transDescr* to describe key transactions, and *activeLoans* to list their current loan obligations. These details form the core data set required for evaluating creditworthiness and are securely recorded in off-chain attestations.

Scorers, functioning as specialized *Verifiers*, are responsible for analyzing the Borrowers' submitted data to generate key credit metrics. The *Scorer* schema includes fields such as *methodology* for documenting the scoring approach, *borrowerAddr* and *borrowerUid* for linking the assessment to the respective Borrower, and *score*, *borrowCapacity*, and *probabilityDefault* for representing the calculated credit metrics. Additional fields, such as *benchmarking* and *financial*, capture the context and details of the scoring process. The *KYB* field (Know Your Business) ensures compliance with network governance, while the *notes* field provides space for qualitative observations.

These fields collectively encapsulate the *Scorer*'s contribution to the evaluation, providing both quantitative and qualitative insights.

Reviewers, another type of *Verifiers*, add a secondary layer of analysis, critically evaluating the metrics produced by Scorers. *Reviewer* attestations include references to the corresponding *Scorer* attestation. This linkage enables verifiability of *Reviewer* adjustments, as they are explicitly tied to the original metrics produced by the Scorer. The *Reviewer* attestation also includes the *Borrower* address in the *Recipient* field. The schema includes fields such as *anchorAdjustment*, *modifiers*, and *totalAdjustment*, which document suggested corrections or refinements to the Scorer's metrics. Additionally, the *probabilityDefaultAdj* field allows Reviewers to offer adjusted probabilities of default, while *keyRisks* identifies critical concerns affecting the Borrower's credit profile. The inclusion of a *notes* field enables Reviewers to provide supplementary remarks or contextual information. The *Reviewer*'s role is crucial in validating and refining the *Scorer*'s outputs, ensuring accuracy and impartiality in the evaluation process. It is important to notice that Reviewers' inputs remain confidential from Borrowers thanks to the off-chain attestation, preserving objectivity and preventing potential conflicts of interest.

The results of individual evaluations — *Scorer* metrics and *Reviewer* adjustments — are aggregated in the *Aggregated Review* schema. This schema consolidates the data into fields such as *aggAtts*, which represents the list of individual off-chain attestation UIDs, and *probabilityDefaultAdj*, which calculates the consensus-adjusted probability of default. The *notes* field allows for the inclusion of overarching observations, while *proofCommitment* relates to the *Proof-of-SQL* hash generated by the decentralized data warehouse. This aggregation step involves calculating the average of *Reviewer* corrections and scores, resulting in a single, consensus-driven on-chain attestation.

7. Evaluation

In this section, we present the evaluation of `VeriNet` for the two selected case studies, focusing on cost and performance metrics. The analysis is structured in three main parts: a granular evaluation of costs, an overall cost analysis, and an assessment of the framework's performance.

The primary focus of our cost analysis is on gas usage and the associated gas fees incurred. To ensure a reliable basis for our analysis, gas fees have been calculated using the average monthly gas cost between October 7th and November 7th, 2024. During this period, the average gas price was determined to be 16.60734375 *Gwei*.

The cost analysis presented in this section is based on the Ethereum Sepolia testnet. All smart contracts and operations were deployed and executed on the testnet, and the gas consumption values were obtained directly from Etherscan transaction records. The gas usage for each operation was then converted to ETH using the average gas price observed during the evaluation period. This methodology was applied to both contract deployment operations and all recurring transactions. The operations are categorized into one-time and recurring transactions for improved readability; further details on these categories will be provided in the following subsection.

Type of Operation	Gas Usage	Gas Fees (ETH)
One-Time Transactions		
Content Creator Resolver Contract Deployment	1,653,237	0.02746
Analyzer Resolver Contract Deployment	2,955,162	0.04908
Content Creator Schema Recording	184,605	0.00307
Analyzer Schema Recording	160,735	0.00266
Aggregated Review Schema Recording	207,339	0.00344
Content Creator Registration	92,078	0.00153
Analyzer Registration	74,910	0.00124
	Total	
	5,328,066	**0.08848**
Recurring Transactions		
Off-chain Content Creator Attestation	46,179	0.00077
Off-chain Analyzer Attestation	46,179	0.00077
Token Reward	57,546	0.00096
Aggregated Attestation	396,659	0.00659
	Total	
	546,563	**0.00909**

Table 5: Breakdown of costs for the Deepfake Detection Case Study

Type of Operation	Gas Usage	Gas Fees (ETH)
One-Time Transactions		
Borrower Resolver Contract Deployment	1,653,237	0.02746
Scorer Resolver Contract Deployment	2,782,168	0.04620
Reviewer Resolver Contract Deployment	3,487,514	0.05791
Borrower Schema Recording	160,578	0.00275
Scorer Schema Recording	230,760	0.00383
Reviewer Schema Recording	184,904	0.00307
Aggregated Review Schema Recording	160,361	0.00266
Borrower Registration	84,124	0.00139
Scorer Registration	88,124	0.00146
Reviewer Registration	77,346	0.00128
	Total	
	8,909,116	**0.14801**
Recurring Transactions		
Off-chain Borrower Attestation	46,179	0.00077
Off-chain Scorer Attestation	46,179	0.00077
Off-chain Reviewer Attestation	46,179	0.00077
Token Reward (Scorer)	57,546	0.00096
Token Reward (Reviewer)	57,546	0.00096
Aggregated Attestation	621,473	0.01032
	Total	
	875,102	**0.01455**

Table 6: Breakdown of costs for the Fintech Credit Scoring Case Study

7.1. Granular Cost Analysis

The granular evaluation of costs delves into the specifics of the gas usage incurred by different operations within `VeriNet`. Tables 5 and 6 provide a summary of these costs, outlining the types of operations, gas consumption, and corresponding gas fees in ETH. The analysis shown in these tables is based on a simplified scenario involving only one *Contributor* and one *Verifier*, depending on the scenario. The sum of the costs in the table refers to a single job, which represents a complete "cycle" or operation of the framework.

The tables are organized to differentiate between *one-time* transactions and *recurring* transactions. One-time transactions refer to operations that are performed only once during the setup phase, such as contract deployments or participant registrations. Recurring transactions, on the other hand, refer to processes that are periodically executed during the use of the framework. Figure 4 shows for both case studies the gas usage for key groups of operations, highlighting the most and least expansive processes.

Regarding the deepfake detection case study and the one-time transactions, as expected the most expensive operations are related to the contract deployment. Specifically, the Analyzer Contract Deployment is more expensive than the Content Creator Contract Deployment. This difference arises because

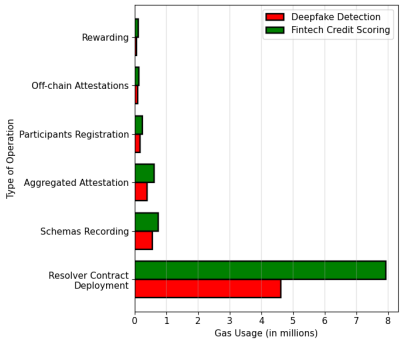

Figure 4: Gas usage comparison based on the type of operation.

the Content Creator contract is primarily responsible for verifying whether participants are included in the framework to participate, whereas the Analyzer contract is more complex: it manages the rewarding mechanism and policy, which depends on factors such as analyzer scores and historical data. Schema recordings constitute the second most expensive category in the table. Their cost depends on the fields defined in the schema and the type of each field. However, the costs of these operations are relatively similar. In contrast, the registration of participants, in this case, the registration of the content creator and the analyzer, represents the least expensive operation. Overall, the one-time transactions account for the most significant expenses, with the majority of the costs stemming from contract deployments.

For the recurring transactions, the costs associated with off-chain attestations are considered for both content creators and analyzers. The off-chain attestation consumes gas — despite its off-chain nature — because its timestamp is anchored on-chain. Hence, this operation always uses the same low-level of gas, regardless of the attestation size.

Beyond this, the token reward phase is more expensive than the timestamping of off-chain attestations. The aggregated attestation process can also be costly, as it depends on the volume and complexity of the fields being processed. This includes the number of attestations, their structure, and the type of data involved, which can require additional computational resources. Moreover, using a DDW introduces operational costs. These include fees for storing off-chain attestations, querying the data to compute metrics, and generating the *Proof-of-SQL*. The DDW we utilize is SpaceandTime, which employs a gas-like approach for its operations. Queries generally cost around 0.00024 ETH, while the creation of a single *Proof-of-SQL* is approximately 0.0016 ETH.

In the Fintech Credit Scoring case study, there are three types of participants. Consequently, in the one-time transaction, we need to consider the deployment of three separate contracts, which results in higher deployment costs compared to the previous case study. There are three schemas to record, each corresponding to one of the entities involved in the case study. While the gas usage across the schemas is similar, the scorer schema is notably the most expensive. This is due to its larger number of fields compared to the borrower and reviewer schemas, as well as the increased complexity of the field types. For example, the borrower and reviewer schemas primarily consist of string types, while the scorer schema includes a wider variety of types, increasing its complexity and, consequently, the gas usage. For the Recurring Transaction, the cost of the off-chain timestamping operation remains the same of the one observed for the deepfake detection case study. This consistency also applies to the token rewards allocated to the scorer and the reviewer. The primary difference between the recurring transactions in the deepfake detection case study and the fintech credit scoring case study lies in the aggregated attestation phase. In this case study, the aggregated attestation phase incurs higher gas usage compared to the deepfake detection scenario. This is because the aggregation process must handle a greater volume of data, driven by the increased complexity of the schemas analyzed earlier.

7.2. Overall Cost Analysis

Following the detailed cost analysis, we present an overall cost evaluation that consolidates these insights for a varying size of the participants' set and attestations. This involves examining how total costs are affected by factors such as attestation data size and the number of *Verifiers*, whose roles vary across the two case studies. This allows us to understand the framework's scalability. Building upon the granular cost measurements, we developed comprehensive evaluation scenarios that examine how attestation data size and the number of verifiers impact overall system costs. To generate the cost curves presented in Figures 5 and 6, we first calculated the total cost for each scenario by summing the one-time transaction costs with the recurring transaction fees in ETH. Subsequently, we varied key parameters, including attestation data size (ranging from 5KB to 100KB) and the number of verifiers (analyzers for deepfake detection, scorers and reviewers for fintech credit scoring) to observe how these variations affect the overall operational expenses. The analysis demonstrates how larger attestations require more gas for processing and storage, directly impacting the total system costs.

In a general cost analysis, particular attention is given to the size of the attestation data. The attestation may contain a large volume of data, and when this data is stored, it results in higher gas consumption. This is an important consideration for understanding the overall costs involved in the process.

To estimate gas usage per byte, we used multiple attestations as reference points. The hexadecimal values of the attestation data were converted to bytes, and based on the gas usage of the transactions involved, we calculated the cost per byte. The gas usage per byte helps us understand how the framework scales as the complexity increases. Figures 5 and 6 provide an overview of the overall costs for the Deepfake Detection and Fintech Credit Scoring case studies, respectively.

The graphs show four distinct curves representing different attestation data sizes: 5, 10, 20, 50, and 100 KB. Additionally, the curves vary based on the number of *Verifiers*, analyzed at 1, 5, 10, 20, 50, and 100. The two figures compare the overall costs in ETH for both case studies. Both scenarios show that costs increase as the number of *Verifiers* grows, with higher attestation

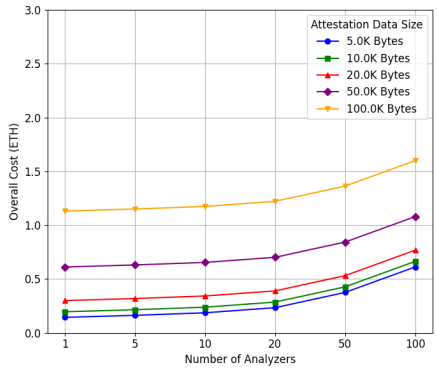

Figure 5: Overall Costs — Deepfake Detection Case Study.

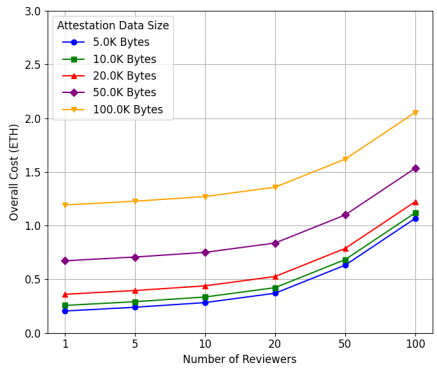

Figure 6: Overall Costs — Fintech Credit Scoring Case Study.

data sizes leading to more significant costs. The Deepfake Detection framework handles analyzers, while the Fintech Credit Scoring case involves more complex roles, including two types of *Verifiers*, such as scorers and reviewers resulting in higher costs. In both cases, larger attestations, such as the 100000-byte data attestation, incur the highest costs due to their complexity and additional computational requirements.

These figures demonstrate how both the size of the attestation and the number of *Verifiers* influence the overall costs of the system. While both case studies show that the framework is cheap at lower participant levels, the costs rise as the number of participants and the size of the attestations increase.

7.3. Scalability Evaluation

This section evaluates the scalability of the `VeriNet` workflow by measuring the execution time of its procedures.

Regardless of the case study, the total time required to execute *Contributor*'s functions is mostly given by:

$$T_C = t_O^{\text{off}} + t_{\mathcal{DDW}}$$

The first term is the off-chain attestation time, while the second term is the DDW storage time. T_C strongly depends on the size of the data to attest.

The total time required to obtain the final attestation recipe is given by:

$$T_{\mathcal{V}} = t_{\mathcal{V}}^{\text{Comp}} + t_O^{\text{Off}} + t_{\mathcal{Q}} + t_{\mathcal{P}} + t_O^{\text{on}}$$

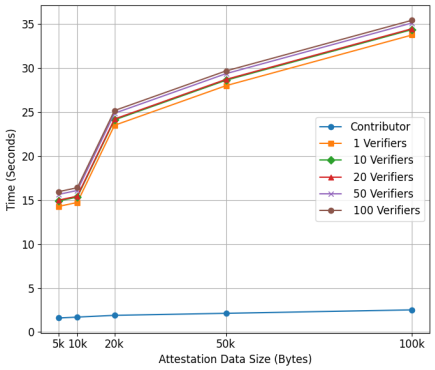

Figure 7: Execution Time for Participants Procedures.

The term $t_{\mathcal{V}}^{\text{Comp}}$ represents the time required for the *Verifier* computation, which, for simplicity, is assumed to be zero as it significantly depends on the context of usage. The second term, t_O^{Off}, corresponds to the time taken to generate the verifier's off-chain attestation. Following this, $t_{\mathcal{Q}}$ denotes the time needed to perform the aggregating query, including the *Proof-of-SQL*. The term $t_{\mathcal{P}}$ captures the time required for processing intermediate steps or additional calculations that may be involved. Finally, t_O^{on} refers to the time required to create the aggregated on-chain attestation. $T_{\mathcal{V}}$ depends on the size of the data going in the off-chain/on-chain attestations and on the number of participating *Verifiers*, which has an impact on the aggregation and proof generation.

Figure 7 shows the execution time for participant procedures as a function of the attestation data size. For the *Contributor*, the execution time remains almost constant regardless of the data size, indicating that its computational load is not influenced by the size of the data. In contrast, the *Verifiers*' execution time increases with both the attestation data size and the number of *Verifiers* involved. For smaller data sizes, the increase in execution time for the *Verifiers* is less visible but becomes more pronounced as the data size grows. This suggests that while the number of *Verifiers* has a limited impact on execution time, the attestation data size is the primary factor influencing it.

8. Discussion

In this section, we address Key Questions (KQs) regarding the operational and security aspects of `VeriNet`. Specifically, we discuss the role of the host, the mechanisms to ensure the honesty of the verifiers, and the decentralized signing process for aggregated attestations.

KQ1: *Who sets the framework infrastructure up?* We envisage the presence of an entity responsible for hosting the framework infrastructure. This entity is tasked with deploying resolver smart contracts, registering schemas on the EAS, and managing the DDW. However, its role does not introduce a trust dependency in the system. Participants in `VeriNet` are not required to trust this entity because the resolver smart contracts are assumed to be open source and verifiable, EAS schemas are not security-critical, and

the DDW is designed to store data in an encrypted form while supporting proof-of-SQL queries to ensure correct data management. This guarantees that while an organization hosts certain infrastructural components, the integrity and security of the framework remain independent of its trustworthiness.

KQ2: *Who guarantees that the verifiers are honest?* The integrity of the verifiers in `VeriNet` is ensured through a combination of incentives and penalties. In addition to the rewarding mechanism, which encourages active and honest participation in the verification process, the framework also implements a slashing mechanism to penalize dishonest behavior. Verifiers are required to stake assets, which serves as a guarantee to act honestly. This staking mechanism operates similarly to a Proof-of-Stake, as described in previous sections. If a verifier provides malicious or misleading attestations, a portion of their staked assets may be slashed. This can be used as a disincentive for dishonest behavior.

KQ3: *Who signs the aggregated on-chain attestations?* The aggregated on-chain attestations are not signed using the host's wallet. To ensure decentralization, a multi-signature mechanism is used, thus allowing multiple participants to collectively sign transactions. The host is only responsible for the setup of the framework but does not have any control over the signing process of the aggregated attestations. This approaches ensures that no single entity has full control over critical operations. In `VeriNet`, it is useful for signing the aggregated attestations in a decentralized manner. Since there are N off-chain attestations corresponding to the various scores provided by different verifiers, the aggregated on-chain attestation must reflect this distributed evaluation process. By requiring N signatures from verifiers, the system guarantees that the final attestation accurately expresses the collective evaluation.

KQ4: *What happens when a Contributor submits false or manipulated content?* When a Contributor submits false or manipulated content to the `VeriNet` framework, the system relies on Verifiers and their detection algorithms to evaluate the submission. Verifiers apply their respective algorithms to analyze the content. If the content exhibits characteristics indicating manipulation or falsification, the Verifiers will generate off-chain attestations containing low confidence scores or negative assessment metrics, depending on the specific evaluation criteria established for the given application domain. The aggregation process subsequently combines these individual assessments. When multiple Verifiers identify the content as false or manipulated, the resulting aggregated score will reflect this collective evaluation. The final on-chain attestation, created through the aggregation process and linked to the Contributor's address via the recipient field, will contain metrics indicating low authenticity or high manipulation probability. The Contributor receives this assessment through the blockchain-based attestation system, providing transparent feedback on their submission.

9. Conclusion

This paper presented `VeriNet`, a decentralized framework designed to address the challenges of content verification in heterogeneous domains. Leveraging blockchain technology and the EAS, VeriNet integrates on-chain and off-chain attestations to ensure privacy, transparency, and immutability. The use of a DDW facilitates provable data aggregation through cryptographic Proof-of-SQL, thus enhancing the trustworthiness of the verification process. Two Proof-of-Concept implementations, dealing with deepfake detection and fintech credit scoring, demonstrate the adaptability of `VeriNet` to diverse application areas. The evaluation campaign results prove its performance and cost-efficiency, showing that the framework can scale to support complex verification tasks while maintaining reliability and transparency.

Acknowledgment

This work has been partially funded by the European Union - Next-GenerationEU - National Recovery and Resilience Plan (NRRP) – MISSION 4 COMPONENT 2, INVESTMENT N. 1.1, CALL PRIN 2022 D.D. 1409 14-09-2022 – (DOSSIER - aDvanced mOnitoring SyStem wIth Enhanced secuRity) CUP I53D23006080001.

References

[1] B. Jacobs, The authenticity crisis, Computer Law & Security Review 53 (2024) 105962.

[2] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, et al., The science of fake news, Science 359 (6380) (2018) 1094–1096.

[3] M. Westerlund, The emergence of deepfake technology: A review, Technology innovation management review 9 (11) (2019).

[4] E. L. Jenkins, J. Ilicic, A. M. Barklamb, T. A. McCaffrey, Assessing the credibility and authenticity of social media content for applications in health communication: scoping review, Journal of medical Internet research 22 (7) (2020) e17296.

[5] Á. Figueira, L. Oliveira, The current state of fake news: challenges and opportunities, Procedia computer science 121 (2017) 817–825.

[6] S. P. Sethi, T. F. Martell, M. Demir, Enhancing the role and effectiveness of corporate social responsibility (csr) reports: The missing element of content verification and integrity assurance, Journal of business ethics 144 (2017) 59–82.

[7] J. Kietzmann, A. J. Mills, K. Plangger, Deepfakes: perspectives on the future "reality" of advertising and branding, International Journal of Advertising 40 (3) (2021) 473–485.

[8] E. Lundberg, P. Mozelius, The potential effects of deepfakes on news media and entertainment, AI & SOCIETY (2024) 1–12.

[9] C. Vaccari, A. Chadwick, Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news, Social media+ society 6 (1) (2020) 2056305120903408.

[10] R. Aissani, R. A.-Q. Abdallah, S. Taha, M. N. Al Adwan, Artificial intelligence tools in media and journalism: Roles and concerns, in: 2023 international conference on multimedia computing, networking and applications (MCNA), IEEE, 2023, pp. 19–26.

[11] A. Alamsyah, G. N. W. Kusuma, D. P. Ramadhani, A review on decentralized finance ecosystems, Future Internet 16 (3) (2024) 76.

[12] N. Xia, X. Zhao, Y. Yang, Y. Li, Y. Li, Exploration on real world assets and tokenization, arXiv preprint arXiv:2503.01111 (2025).
[13] H. He, Z. Wang, H. Jain, C. Jiang, S. Yang, A privacy-preserving decentralized credit scoring method based on multi-party information, Decision Support Systems 166 (2023) 113910.
[14] M. Tigges, S. Mestwerdt, S. Tschirner, R. Mauer, Who gets the money? a qualitative analysis of fintech lending and credit scoring through the adoption of ai and alternative data, Technological Forecasting and Social Change 205 (2024) 123491.
[15] Ethereum Attestation Service (EAS), Ethereum attestation service documentation (2024).
URL `https://docs.attest.org/`
[16] B. Boi, C. Esposito, J. T. Seo, Ethereum attestation service as a solution for the revocation of hardware-based password-less mechanisms, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, 2024, pp. 553–559.
[17] Proof-Of-SQL, Github repository of sxt proof-of-sql (2024).
URL `https://github.com/spaceandtimelabs/sxt-proof-of-sql`
[18] S. Dykstra, J. White, N. Holiday, C. Daly, D. Alves, I. Joiner, Space and time (2024).
[19] J. E. Uscinski, R. W. Butler, The epistemology of fact checking, Critical Review 25 (2) (2013) 162–180.
[20] Y. Zheng, D. Wu, The impact of opacity on bank valuation during the global financial crisis: A channel analysis, International Review of Financial Analysis 87 (2023) 102580.
[21] H. Gao, J. Wang, X. Yang, L. Zhao, Borrower opacity and loan performance: evidence from china, Journal of Financial Services Research 57 (2) (2020) 181–206.
[22] F. Härer, H.-G. Fill, Decentralized attestation of conceptual models using the ethereum blockchain, in: 2019 IEEE 21st Conference on Business Informatics (CBI), Vol. 1, IEEE, 2019, pp. 104–113.
[23] F. Härer, H.-G. Fill, Decentralized attestation and distribution of information using blockchains and multi-protocol storage, IEEE Access 10 (2022) 18035–18054.
[24] S. He, X. Xing, G. Wang, Z. Sun, A data integrity verification scheme for centralized database using smart contract and game theory, IEEE Access 11 (2023) 59675–59687.
[25] X. Zhou, A. Nehme, V. Jesus, Y. Wang, M. Josephs, K. Mahbub, A. Abdallah, Audiwflow: Confidential, collusion-resistant auditing of distributed workflows, Blockchain: Research and Applications 3 (3) (2022) 100073.
[26] S. Paul, J. I. Joy, S. Sarker, S. Ahmed, A. K. Das, et al., Fake news detection in social media using blockchain, in: 2019 7th international conference on smart computing & communications (ICSCC), IEEE, 2019, pp. 1–5.
[27] C. N. Buțincu, A. Alexandrescu, Blockchain-based platform to fight disinformation using crowd wisdom and artificial intelligence, Applied Sciences 13 (10) (2023) 6088.
[28] H. R. Hasan, K. Salah, Combating deepfake videos using blockchain and smart contracts, Ieee Access 7 (2019) 41596–41606.
[29] J. A. Costales, S. Shiromani, M. Devaraj, The impact of blockchain technology to protect image and video integrity from identity theft using deepfake analyzer, in: 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), IEEE, 2023, pp. 730–733.
[30] M. Priya, J. Murugesan, P. Bhuvaneswari, M. Rubigha, S. Lalithambikai, B. Mohanraj, Preserving visual authenticity: Block chain-augmented ai frameworks for advanced digital deception recognition and mitigation, in: 2024 5th International Conference on Smart Electronics and Communication (ICOSEC), IEEE, 2024, pp. 707–713.
[31] K. Bindra, H. Gupta, D. Bhattacharya, R. Thakur, Hyperswin: preventing deepfake proliferation with swin-efficient fusion in a hyperledger ecosystem, Signal, Image and Video Processing 19 (8) (2025) 1–14.
[32] J. Zhang, R. Tan, C. Su, W. Si, Design and application of a personal credit information sharing platform based on consortium blockchain, Journal of Information Security and Applications 55 (2020) 102659.
[33] V. Hassija, G. Bansal, V. Chamola, N. Kumar, M. Guizani, Secure lending: Blockchain and prospect theory-based decentralized credit scoring model, IEEE Transactions on Network Science and Engineering 7 (4) (2020) 2566–2575.
[34] Z. Jovanovic, Z. Hou, K. Biswas, V. Muthukkumarasamy, Robust integration of blockchain and explainable federated learning for automated credit scoring, Computer Networks 243 (2024) 110303.
[35] J. S. Hunter, The exponentially weighted moving average, Journal of quality technology 18 (4) (1986) 203–210.