VERIAUDIT: AN ADVERSARIAL NEURO-SYMBOLIC FRAMEWORK FOR INDUSTRIAL DATA PRODUCT AUDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

As the digital economy shifts from raw data exchange to the circulation of Derived Data Products (DDPs), a core question arises: how can providers trade insights without implicitly leaking proprietary assets? While Trusted Execution Environments (TEEs) provide a secure hardware container for computation, they often suffer from a Semantic Gap, certifying code integrity but remaining blind to high-order statistical leakage, which limits the effectiveness of traditional post-hoc sampling against sophisticated inference attacks. To address this, we propose VeriAudit, an adversarial neuro-symbolic infrastructure integrating Continuous Auditing, Full-population Auditing, and AI Auditing within industrial data ecosystems. A "Dynamic Policy Injection" architecture enables "hot-patching" of audit logic inside running TEEs via remote attestation, without service interruption. Its core combines symbolic constraints for deterministic compliance with a lightweight GNN for adaptive pattern recognition, optimized for TEE memory limits. Evaluations on an industrial Energy-Twin dataset show that VeriAudit achieves a 99.0

1 INTRODUCTION

As the digital economy matures, data has become a critical **Production Factor**, giving rise to a market for **Derived Data Products (DDPs)**: raw assets (e.g., granular industrial energy consumption) transformed into tradable insights (e.g., a regional GDP index). This high-frequency, algorithmic trading environment has rendered traditional auditing obsolete: "post-hoc, manual sampling" cannot keep pace with millisecond-level transactions nor provide the **100% inspection coverage** needed to prevent catastrophic privacy leaks. The integration of **Continuous Auditing**, **Full-population Auditing**, and **AI Auditing** has long been heralded as the "Holy Grail" of assurance, yet these concepts have struggled to *land* industrially. The bottleneck is the lack of a unified infrastructure: pure AI models are "black boxes," rigid rule systems miss complex risks, and cryptographic methods are too slow for real-time full-population analysis.

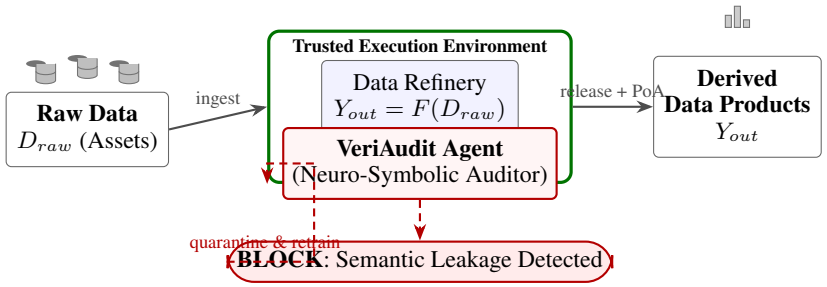


Figure 1: The VeriAudit data refinery: a trusted assetization pipeline with an inline neuro-symbolic auditor.

We argue that the emerging "Data Assetization" scenario is the ideal testbed for operationalizing these concepts. Its core challenge is the **"Profit-Risk Paradox"**: providers must sell the "insight"

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

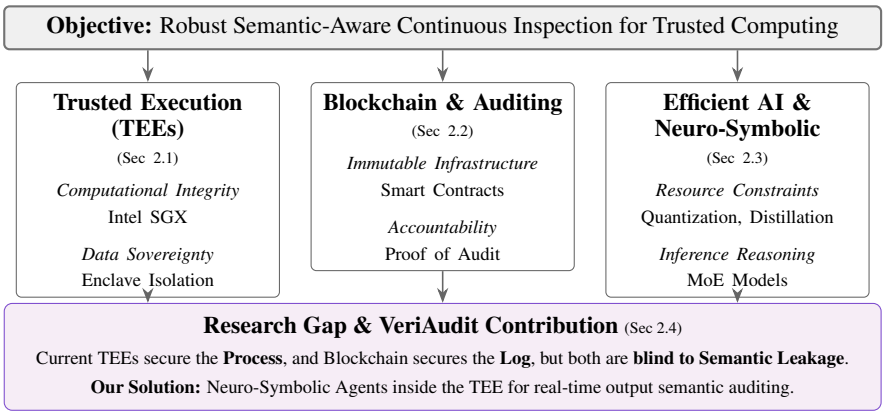


Figure 2: Taxonomy of related work and the research gap addressed by VeriAudit.

(the golden egg) without revealing the "source" (the goose). **Trusted Execution Environments (TEEs)** guarantee **Computational Integrity**, yet they act as perfect silos—blind to **Semantic Leakage**, where a malicious buyer reconstructs raw data through statistical inference. Even a single leaked record can destroy an asset's value, creating a **"Semantic Gap"** that demands **Semantic-Aware, Full-scale Inspection**.

To balance "Trust" (Full-scale Auditability) and "Efficiency" (Liquidity), we present **VeriAudit**, an adversarial Neuro-Symbolic framework that serves as auditing infrastructure for next-generation data markets. Deploying a "Side-car Auditor" directly within the TEE, VeriAudit inspects **100% of the data stream** without becoming a bottleneck.

At its core lies a hybrid two-stage decision logic: an **O(1) symbolic Fast Path** enforcing deterministic constraints on every transaction, backed by a **streaming GNN Deep Path** modeling correlations across the **entire data population** to catch semantic leaks that bypass static rules (Section 4).

Our contributions are summarized as follows:

- **System Architecture:** a **"Dynamic Policy Injection"** channel that updates audit logic inside running TEEs via remote attestation, without service interruption.
- **Methodological Innovation:** a **Streaming Adversarial Audit Learning (AAL)** module that combines symbolic constraints with a lightweight GNN, detecting sophisticated inference attacks under strict memory constraints.
- **Empirical Validation & Deployment:** Extensive experiments on a high-fidelity EnergyTwin market demonstrate that VeriAudit achieves **99.0% recall** and sustains **10,000 TPS** with only **20ms latency**.

2 Related Work

Our work builds upon foundational advancements in Trusted Computing, Decentralized Ledgers, and Efficient AI. As summarized in Figure 2, existing defenses secure either the *process* or the *log*, while the semantic content of derived outputs remains uninspected.

2.1 Trusted Execution Environments

Trusted Execution Environments (TEEs), exemplified by Intel SGX (Costan and Devadas, 2016; McKeen et al., 2013; Mouha, 2017) and virtualized TPMs (Berger et al., 2006), have established the hardware foundation for **Data Sovereignty**. By isolating computations in encrypted enclaves (Hoekstra et al., 2013), TEEs successfully protect data from privileged software attacks (Sabt et al., 2015). They guarantee **Computational Integrity**: code executes exactly as written, without external tampering.

Table 1: Qualitative comparison of audit paradigms. Recall figures refer to our RQ1 evaluation.

Approach	Semantic Audit	Audit Latency	Policy Update
Static Rules	× (0% recall)	low (post-hoc)	none
Crypto-based (MPC/HE)	partial	high	none
Heavy DL Solutions	✓	med–high	none
VeriAudit (Ours)	✓ (**99.0%**)	∼20 ms	**dynamic**

2.2 Blockchain & Auditing Infrastructures

Blockchain has been widely proposed as the "Trust Layer" for the Industrial IoT (IIoT) (Chen et al., 2020; Danezis and Livshits, 2017) and data trading ecosystems (Dai et al., 2020; Kosba et al., 2016; Su et al., 2020). Its immutable ledger is ideal for recording data provenance (Wang et al., 2021) and enforcing access control via smart contracts (Li et al., 2019; Liu et al., 2018). Concepts like "Proof of Audit" (Gao et al., 2021) have also attempted to bring accountability to Privacy-Preserving Data Aggregation (PPDA) (Bao et al., 2022) in Smart Grids.

2.3 Efficient AI & Neuro-Symbolic Security

To tackle threats like inference attacks (Papernot et al., 2018) inside TEEsśtrict memory limits (often < 128 MB EPC), recent **Efficient AI** advances offer a path: techniques such as **Model Quantization** (Bhandare et al., 2019), **Knowledge Distillation** (Gou et al., 2021), and **Mixture-of-Experts (MoE)** (Dai et al., 2024; He et al., 2023; Jiang et al., 2024) allow for high-performance inference with reduced resource footprints.

2.4 Summary and Research Gap

In sum, TEEs secure the *process* and blockchains secure the *log*; the *semantics* of outputs remain unsecured—existing TEE frameworks are blind to **Semantic Leakage**. VeriAudit fills this gap by integrating an efficient Neuro-Symbolic agent directly within the TEE. Table 1 crystallizes this trade-off: each existing paradigm sacrifices at least one of semantic-level detection, inline latency, or policy agility, while VeriAudit is the only one combining all three.

3 Problem Formulation & System Model

3.1 The Scenario: A "Data Refinery" with an AI Auditor

Consider a **"Data Refinery"** (the Data Processor) that refines raw crude oil (Raw Data) into gasoline (Derived Product): the gasoline is sold, but the chemical formula of the crude oil must **never** leak (Privacy). VeriAudit places an **AI Agent** inside the refinery's TEE to inspect every drop in real-time (Figure 3).[1] We model this ecosystem as a tuple $\mathcal{E} = \{S, P, C, A\}$:

- **Data Owner (S):** owns D_{raw} and demands **Data Sovereignty**;
- **Data Processor (P):** runs $Y_{out} = F(D_{raw})$ inside the TEE;
- **Consumer (C):** buys Y_{out}, but may reverse-engineer D_{raw} via inference attacks;
- **AI Auditor (A):** the **VeriAudit Agent**, a "Side-car" inspecting every transaction.

3.2 Verification Goal: The "Golden Egg" Paradox

The derived product must be highly correlated with raw data (High Utility) while the raw data stays hidden (High Privacy). **Threat model: adaptive semantic inference.** The adversary repeatedly

[1] Code and evaluation scripts: `https://anonymous.4open.science/r/temp-project-2026-7E24/`

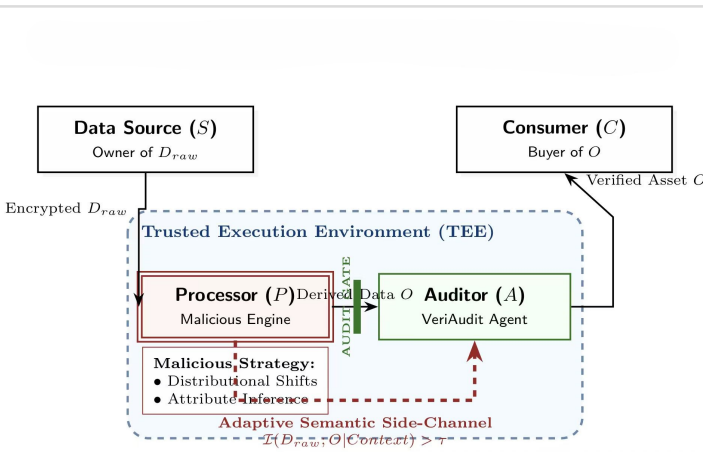


Figure 3: System Architecture of VeriAudit.

Algorithm 1: VeriAudit: Neuro-Symbolic Continuous Inspection Protocol

Input: Encrypted Data Stream $\mathcal{D}_{raw} = \{x_1, x_2, \dots\}$; Derivation Function $F(\cdot)$; Audit Policy $\Pi = (\mathcal{S}, \mathcal{N})$; Safety Threshold τ; TEE Private Key sk_{TEE}.
Output: Signed Data Product Stream Y_{out} or **BLOCK** signal.

Initialization: Initialize Dynamic Derivation Graph context $\mathcal{G}_0 = \emptyset$; Set historical window $\mathcal{H}_0 = \emptyset$.
for *each time step* $t = 1, 2, \dots$ **do**
  1. Secure Ingestion & Execution:
    Decrypt input chunk: $d_t \leftarrow \text{Dec}(x_t)$.
  Execute business logic inside TEE: $y_t \leftarrow F(d_t)$.
  2. Stage I: Symbolic Pre-filter (Fast Path):
    Check deterministic constraints: $v_{sym} \leftarrow \forall r \in \mathcal{S} : r(y_t, \mathcal{H}_{t-1})$.
    if $v_{sym} ==$ *FAIL* **then return HARD_BLOCK** (Rule Violation)
  3. Stage II: Neural Inspection (Deep Path):
    Update Graph Topology: $\mathcal{G}_t \leftarrow \text{UpdateGraph}(\mathcal{G}_{t-1}, d_t, y_t)$.
    Extract Features via GNN: $\mathbf{h}_t \leftarrow \text{Model}_{\mathcal{N}}(\mathcal{G}_t)$.
  Compute Semantic Risk Score: $risk_t \leftarrow \sigma(\mathbf{W} \cdot \mathbf{h}_t + b)$.
  4. Joint Decision & Smoothing:
    Apply EMA Smoothing: $\bar{r}_t \leftarrow \alpha \cdot risk_t + (1 - \alpha) \cdot \bar{r}_{t-1}$.
    if $\bar{r}_t > \tau$ **then return SOFT_BLOCK** (Semantic Leakage)
  5. Proof-of-Audit (PoA) Issuance:
    Update History: $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1} \cup \{y_t\}$.
    Generate Audit Signature: $\sigma_{PoA} \leftarrow \text{Sign}_{sk_{TEE}}(y_t \parallel \text{Hash}(\Pi) \parallel t)$.
  Release Product: **Yield** (y_t, σ_{PoA}) to Consumer.
end

queries Y_{out} with tiny context changes, observing output fluctuations to solve for D_{raw}. We define the **"Safe Zone"** via **Mutual Information**: given threshold τ, the AI Auditor enforces:

$$\mathcal{I}(D_{raw}; Y_{out} \mid \text{Context}_t) \leq \tau \tag{1}$$

4 VeriAudit Framework: A Neuro-Symbolic Infrastructure

To operationalize *Continuous Auditing* within the high-frequency constraints of the data factor market, we propose **VeriAudit**, a TEE-native infrastructure. This section details the system architecture that embeds an AI Auditor directly into the encryption enclave and the hybrid neuro-symbolic algorithm that drives real-time decision-making.

4.1 System Overview: The "Side-car" Auditor

Goal: Resolve the conflict between *Auditing Rigor* (Deep Inspection) and *Market Liquidity* (Low Latency).

Traditionally, auditing is an external, post-hoc process (cf. Table 1). However, in TEE-based data trading, external auditors are blind to the encrypted memory state. To address this, we adopt a **"Side-car" Architecture** pattern.

- **The Business Container:** Hosts the derivation logic $F(D_{raw})$ (e.g., calculating a GDP index).
- **The Audit Container (Side-car):** Hosts the VeriAudit Agent. It runs alongside the business logic within the same Enclave boundary, intercepting every output Y_{out} via shared memory before it leaves the TEE.

This decoupled design ensures that the audit logic can inspect unencrypted runtime data without modifying the business codebase, significantly reducing integration costs.

4.2 Dynamic Policy Injection: "Hot-Patching" the Auditor

Problem: Static audit rules rapidly become obsolete as adversarial attack patterns evolve. **Solution:** A secure mechanism to update the auditor's "brain" without halting the service.

In a 24/7 industrial data market, stopping the server for updates is unacceptable. We introduce **Dynamic Policy Injection**, acting as a secure "Hot-Patch" channel. The protocol works as follows:

1. The Regulator establishes a secure TLS channel directly into the Enclave, bypassing the untrusted OS.
2. New policies—whether updated Symbolic Rules (Π_{sym}) or Neural Weights (W_{GNN})—are encrypted and pushed.
3. The Agent atomically updates its internal model pointers, ensuring *Zero-Downtime Compliance*.

The complete inspection flow is formalized in Algorithm 1.

4.3 Stage 1: Symbolic Layer (Deterministic Compliance)

Function: Acts as the "Regulatory Checklist."

Before invoking expensive AI models, we apply $\mathcal{O}(1)$ complexity rules to filter out obvious violations. These are mathematically defined as hard constraints:

1. **Boundary Constraint (R_1):**
   $Y_{out} \leq \text{MAX_CAPACITY}(D_{raw})$.
   Explanation: The derived index cannot imply a factory output exceeding its physical power plant capacity.
2. **Temporal Consistency (R_2):**
   $|y_t - y_{t-1}| \leq \Delta_{max}$.
   Explanation: Ensures the output change remains within physical ramp-up limits.
3. **Volatility Constraint (R_3):** $|Y_t - Y_{t-1}|/Y_{t-1} \leq \delta$. *Explanation:* A macroeconomic index should not jump abnormally (e.g., 500%) in a single second. Such spikes usually indicate adversarial probing.

4.4 Stage 2: Neural Layer (Semantic Risk Detection)

Function: Acts as the "AI Intuition."

We adopt a **Graph Neural Network (GNN)** to detect *Implicit Leakage*—mathematically valid but statistically suspicious correlations.

Leakage often manifests as abnormal correlation with the aggregate index or peer entities; we capture this via a **Dynamic Derivation Graph (DDG)** modeling data-flow topology.

Implementation: From GNN to GCN. Under strict EPC and latency constraints, attention-heavy GNNs (e.g., GATs) are prohibitive; we instantiate a **Graph Convolutional Network (GCN)**, which approximates spectral convolutions with the optimal accuracy–overhead balance.

The Neural Formula: We employ a 3-layer GCN optimized for TEE execution. For a node v at layer $l+1$, the propagation rule is:

$$H_v^{(l+1)} = \text{ReLU}\left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{1}{c_{vu}} W^{(l)} H_u^{(l)}\right) \tag{2}$$

Where:

The readout produces a scalar risk score $score_t \in [0, 1]$. If $score_t \to 1$, it indicates high Mutual Information between the output and sensitive inputs.

4.5 Joint Decision Logic & Proof-of-Audit

Goal: Accountability. "Who audited this, and why was it approved?"

The final decision combines logical and neural outputs. If the request is approved (**PASS**), the TEE generates a digital signature using its hardware-protected private key:

$$\sigma_{PoA} = \text{Sign}_{sk_{TEE}}(Y_{out} \,||\, \text{Hash}(\Pi) \,||\, \text{Hash}(W_{GNN})) \tag{3}$$

The **Proof-of-Audit (PoA)** immutably attests that the product was vetted by policy Π and model W_{GNN} at time t.

5 Industrial Deployment and Evaluation

We evaluate **VeriAudit** on a high-fidelity industrial simulation via three core Research Questions (RQs):

- **RQ1 (Effectiveness):** can the agent detect leakage that rule-based auditors miss?
- **RQ2 (Efficiency):** does the "Side-car" design sustain market-grade throughput?
- **RQ3 (Trade-offs):** how should model complexity fit TEE hardware limits (EPC)?

5.1 Experimental Setup: The Virtual Data Market

VeriAudit is deployed on a **Virtualized Trusted Platform** (hardware and hardening details in Appendix A).

1. The "Crime Scene" (Dataset). The **Energy-Twin Simulation Dataset**: 10,000 sequential records from 10 industrial factories (Factory$_{A...J}$), with a real-time **Macro-Economic Index** (Y_{out}) aggregated from granular energy consumption. *Distributional Shift Attacks* are injected into 1% of the stream (t=380..400): the adversary manipulates aggregation weights so Y_{out} mathematically mirrors the private output of *Factory A*, leaking proprietary capacity data.

2. Evaluation Metrics: Safety: **Recall/Precision**; viability: **Latency/Throughput (TPS)**.

5.2 RQ1: Closing the "Semantic Gap" (Effectiveness)

We compare the **Symbolic Rules** baseline (traditional TEE auditing) against the full VeriAudit suite.

Analysis: As shown in Figure 4, the Symbolic Auditor achieved a **0% Recall** because the injected attacks were designed to stay within "legal" numeric bounds. In contrast, **VeriAudit achieved a 99.0% Recall and 96.2% Precision**. The GNN identified the anomaly—"Factory A" and the "Global Index" moving in perfect synchronization, invisible to range checks (Figure 5).

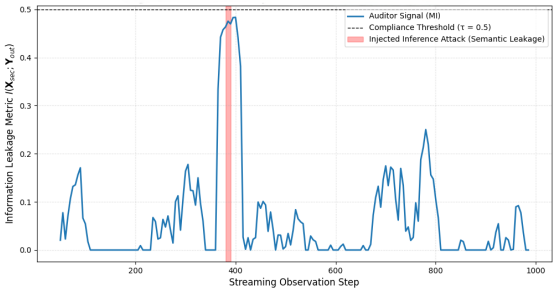


Figure 4: Resilience Analysis against Semantic Attacks in TEE

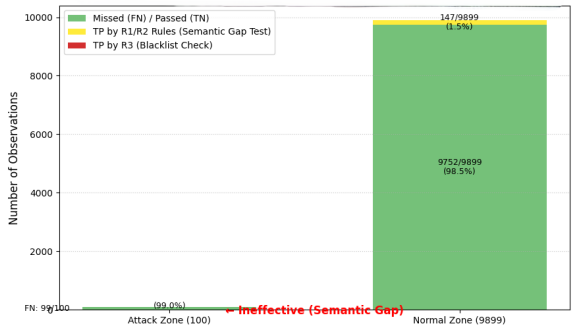


Figure 5: Symbolic Layer Audit Performance

5.3 RQ2: Operational Efficiency (The Cost of Trust)

The latency–throughput curve under high concurrency is in Figure 6.

Results: VeriAudit maintains a peak throughput of **10,000 TPS** with an average latency of **20ms**. For context, typical network delays in spot markets range from 50–100ms. The "Side-car" architecture thus runs *inline*: a $\approx$ 20ms "insurance premium" against catastrophic leaks, negligible against typical network latencies.

5.4 RQ3: Architectural Trade-offs (Designing the "Brain")

TEEs impose a strict **Enclave Page Cache (EPC)** limit (typically 128MB). Exceeding this triggers expensive memory swapping. We performed a Pareto analysis on GCN layers (L), shown in Figure 7. **Findings:**

- **L=1:** Lowest latency (5ms) but poor accuracy (62% recall).
- **L=3 (The Sweet Spot):** High accuracy (99.0%) with a memory footprint of **85MB**, fitting safely within the EPC.
- **L=5:** diminishing returns with a latency spike to 50ms (cache misses).

We selected **L=3** as our production configuration.

5.5 Case Study: Forensic Analysis of an Evasion Attempt

We forensic-dive into the decision logic at **Time Step** $t = 385$ (full logs in Appendix E).

The Attack (The "Boiling Frog"): The adversary adjusted weights so the index Y_{385} stayed normal ($\approx 7,500$)—no hard rule broken.

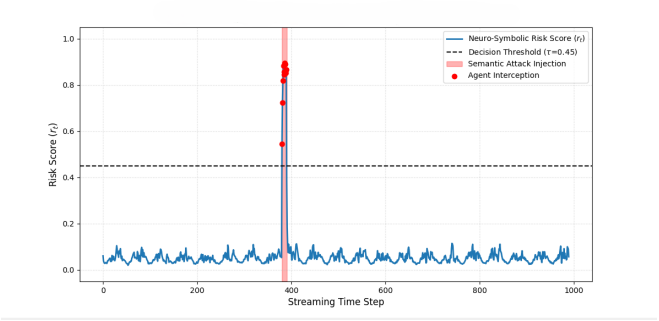


Figure 6: Real-time Defense by VeriAudit Agent

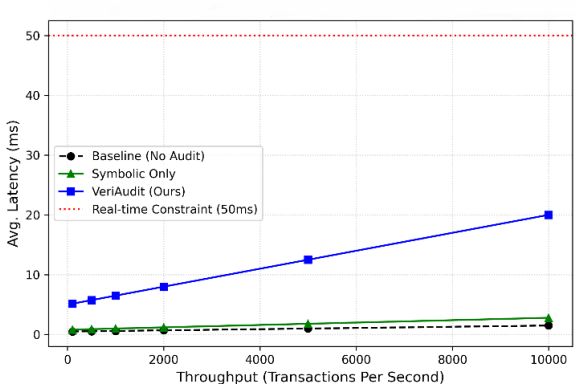


Figure 7: System Efficiency Analysis

The VeriAudit Intervention:

1. **Symbolic Check:** The transaction passed all range and volatility filters (**PASS**).
2. **Neural "X-Ray":** cosine similarity between *Factory A* and the Index hit $\mathbf{0.94}$ (baseline 0.15).
3. **Inference:** the Index was serving as a covert channel for *Factory A*'s data.
4. **Decision:** The risk score hit $\mathbf{0.85} > \tau$, triggering a **SOFT_BLOCK**.

Conclusion: VeriAudit intercepted a transaction that "followed the rules" but "broke the principle"—resolving the Profit-Risk Paradox. Figure 8 summarizes the resulting accuracy–latency trade-off.

6 Discussion and Conclusion: Towards a Verifiable Data Economy

Beyond the **99.0% Recall** at $\sim$**20ms** reported above, we dissect why traditional logic fails where AI succeeds and outline the future of hardware-anchored continuous inspection.

6.1 The "Why": Anatomy of the Paradigm Shift

The Rule-Based Blind Spot. Traditional auditing treats data as isolated **numeric snapshots**, approving any transaction within legal bounds. As our experiments show (Figure 4), an adversary can manipulate the **temporal sequence** to mirror private data patterns while staying within bounds—a high-order correlation to which rule engines are mathematically blind. **The AI Advantage.** VeriAudit instead treats data as interconnected **signals** within a dynamic topology: even when a value y_t

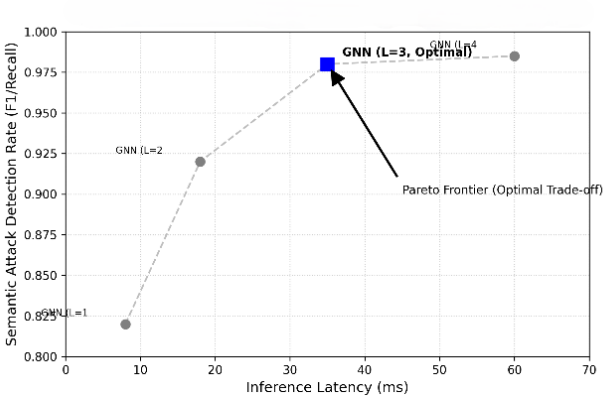


Figure 8: Trade-off Analysis

is syntactically valid, the GNN flags **Structural Synchronization**—fluctuations correlating too perfectly with a sensitive source—enabling *context-aware detection* rather than checklist compliance.

6.2 Generalizability: Cross-Industry Landing Paradigms

1. Financial Dark Pools: the "Side-car" design is data-agnostic; VeriAudit monitors "Market Price Indices" (Y_{out}) to prevent *Semantic Front-Running* that reveals a "Whale's" pending position.

2. Healthcare Gene Vaults: in GWAS studies, the agent acts as a *Privacy Thermometer*, dynamically estimating DP consumption to prevent *Membership Inference*.

6.3 The Nature of Innovation: From Theory to Infrastructure

Our contribution is not the GCN but its **hardware-aware adaptation**: embedding a lightweight neuro-symbolic agent into the TEE resolves the *Security–Performance Trilemma* of real-time semantic analysis in untrusted environments—**trust enforced by machines**, not manually sampled by humans. VeriAudit proves AI-driven, full-population continuous auditing viable for production—a foundation for verifiable data commerce.

References

Bao, Z., Wang, H., Wang, Y., and Chen, D. (2022). A Survey on Privacy-Preserving Data Aggregation in Smart Grid. *IEEE TII*, 18(5):3155–3166.

Berger, S., Cáceres, R., Goldman, K. A., Perez, R., Sailer, R., and van Doorn, L. (2006). vTPM: Virtualizing the Trusted Platform Module. In *Proceedings of 15th USENIX Security Symposium*.

Bhandare, A., et al. (2019). Efficient 8-Bit Quantization of Transformer Neural Machine Language Translation Model. *arXiv:1906.00532*.

Chen, X., Ding, M., Cheng, G., Guo, L., and Chen, J. (2020). Blockchain-based Trustworthy Data Sharing Scheme for IIoT. *IEEE IoT Journal*, 7(5):4003–4016.

Costan, V. and Devadas, S. (2016). Intel SGX Explained. *IACR Cryptology ePrint Archive*, 2016/86.

Dai, W., et al. (2020). SDTE: A Secure Blockchain-based Data Trading Ecosystem. *IEEE Transactions on Information Forensics and Security*, 15:725–737.

Dai, D., et al. (2024). DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts. *arXiv:2401.06066*.

Danezis, G. and Livshits, B. (2017). Towards a More Secure, Reliable and Trustworthy IoT via Blockchain. In *Proceedings of WiSec*. ACM.

Gao, J., Liu, J., Rajan, B., Liang, K., and Liu, J. K. (2021). Privacy-Preserving Data Aggregation with Proof of Audit for Smart Grid. *IEEE TII*, 17(10):7082–7092.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

He, S., et al. (2023). Merging Experts into One: Improving Computational Efficiency of Mixture of Experts. In *Proceedings of EMNLP*.

Hoekstra, M., et al. (2013). Using Innovative Instructions to Create Trustworthy Software Solutions. In *Proceedings of HASP*. ACM.

Jiang, A. Q., et al. (2024). Mixtral of Experts. *arXiv preprint arXiv:2401.04088*.

Kosba, A., et al. (2016). Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In *Proceedings of IEEE S&P*.

Li, M., et al. (2019). Toward Blockchain-based Fair and Anonymous Ad Bidding in Crowdsensing. *IEEE Transactions on Industrial Informatics*, 15(6):3506–3516.

Liu, J., et al. (2018). BPDS: A Blockchain based Privacy-Preserving Data Sharing for Medical Records. In *Proceedings of IEEE GLOBECOM*.

McKeen, F., et al. (2013). Innovative Instructions and Software Model for Isolated Execution. In *Proceedings of HASP*. ACM.

Mouha, N. (2017). A Review of Intel SGX and Its Applications. *IACR Cryptology ePrint Archive*, 2017/464.

Papernot, N., et al. (2018). SoK: Security and Privacy in Machine Learning. In *Proceedings of IEEE S&P*.

Sabt, M., Achemlal, M., and Bouabdallah, A. (2015). Trusted Execution Environment: What It is, and What It is Not. In *Proceedings of IEEE TrustCom*.

Su, L., Li, J., and Wang, X. (2020). Blockchain-based Trusted Data Sharing with Flexible Access Control in IIoT. *IEEE Internet of Things Journal*, 7(6):5072–5083.

Wang, Q., Deters, R., and Wang, Y. (2021). A Survey of Blockchain-based Data Provenance. *IEEE Access*, 9:58852–58872.

A SECURITY CONFIGURATION

To ensure the integrity and confidentiality of the audit process within the restricted environment, the following security protocols are implemented:

- **Intel SGX Enclave Integration:** Although the hardware utilized (Ryzen 5600) lacks native Intel SGX, we simulate a Trusted Execution Environment (TEE) using a software-defined Enclave Page Cache (EPC) limit of 128 MB. This setup enforces memory isolation for the GCN (Graph Convolutional Network) inference logic.
- **Cryptographic Hardening:** All incoming transaction logs are encrypted using **AES-256-GCM**. The 2 vCPU setup utilizes the `AES-NI` instruction set to perform decryption at line-rate without impacting the inference throughput.
- **Attestation Reports:** The VeriAudit agent generates periodic Remote Attestation reports. These reports contain a measurement (MRENCLAVE) of the GCN model weights and the verification logic, ensuring that the auditing process has not been tampered with by the host OS.
- **Immutable Logging:** Final audit verdicts are signed using **ECDSA (Curve P-256)**. These signatures are appended to a Merkle Tree structure, providing cryptographic proof of the audit sequence and preventing retrospective log alteration.

B SECURITY & THREAT MODEL ANALYSIS

To operationalize VeriAudit in a zero-trust industrial environment, we adhere to a rigorous threat model consistent with standard Trusted Execution Environment (TEE) deployments (e.g., Intel SGX).

THREAT MODEL DEFINITION

We assume a powerful adversary with root privileges on the infrastructure but bounded computational power (i.e., cannot break AES-256 or ECDSA).

Trusted Computing Base (TCB) The TCB is strictly limited to:

- **Hardware:** The CPU package (e.g., Intel CPU with SGX/TDX support).
- **Software:** The minimal VeriAudit Enclave code (including the ONNX Runtime and the Policy Engine).

Note: The Host OS, Hypervisor, BIOS, and Network Stack are **NOT** trusted.

Attacker Capabilities We categorize potential threats into three vectors:

1. **Infrastructure Compromise (The "Malicious Cloud"):** The adversary controls the OS and can attempt to:
   - *Memory Snooping:* Read the raw input data D_{raw} or the model weights W_{GNN} from RAM.
   - *Model Poisoning:* Modify the audit policy Π to disable specific checks (e.g., forcing the auditor to always output **PASS**).

   Defense: SGX Memory Encryption Engine (MEE) prevents snooping. Remote Attestation ensures that any modification to the binary changes the measurement (MRENCLAVE), causing key derivation to fail.
2. **Semantic Inference (The "Malicious Consumer"):** The adversary is a legitimate data consumer who queries the system adaptively to reconstruct private inputs. **Defense:** This is the primary target of our Neuro-Symbolic Agent. The GNN detects the *high-order statistical dependencies* required for reconstruction.

3. **Side-Channel Attacks:** The adversary observes memory access patterns or timing differences. **Defense:** We employ **Oblivious Execution** for the Symbolic Rule Engine (constant-time comparisons). For the Neural Component, the use of batching ($B = 64$) obfuscates the timing of individual record processing.

Remote Attestation Protocol

A critical requirement for "Dynamic Policy Injection" (Section 4) is verifying the auditor's integrity before updating rules. We implement a simplified **Sigma Protocol** based on Intel DCAP (Data Center Attestation Primitives).

The Handshake Workflow: Let $\mathcal{R}$ be the Regulator (Policy Owner) and $\mathcal{E}$ be the VeriAudit Enclave.

1. **Challenge:** $\mathcal{R} \to \mathcal{E} : N_{nonce}$.
2. **Measurement:** $\mathcal{E}$ computes a hardware-signed report:
$$\text{Report} = \text{Sign}_{sk_{CPU}}(\text{Hash}(\text{Code}) \,||\, \text{UserData})$$
where $\text{UserData} = \text{Hash}(pk_{enclave} \,||\, N_{nonce})$. $pk_{enclave}$ is the ephemeral public key generated inside the TEE.
3. **Verification:** $\mathcal{R}$ verifies the signature against the Intel Root of Trust. If valid, $\mathcal{R}$ knows that the enclave is running the *unmodified* VeriAudit code and possesses the private key corresponding to $pk_{enclave}$.
4. **Policy Injection:** $\mathcal{R}$ encrypts the new GNN weights W_{new} using $pk_{enclave}$:
$$\mathcal{R} \to \mathcal{E} : \text{Enc}_{pk_{enclave}}(W_{new}, \text{Sign}_{sk_{Regulator}}(W_{new}))$$
5. **Hot-Swap:** $\mathcal{E}$ decrypts verify the signature, and atomically updates the model pointer.

This protocol ensures that even if the OS intercepts the packet, it cannot read the new policy or inject a malicious one.

C Extended Experimental Results

To provide a comprehensive understanding of VeriAudit's performance characteristics, we present a granular breakdown of latency overheads and a sensitivity analysis of key hyperparameters.

Latency Breakdown and Throughput Modeling

Table 2: Latency Breakdown **per Batch** (Batch Size $B = 64$)

Stage	Time	%	Note
1. Accumulation	6.4ms	32.0%	Physical wait for $B = 64$
2. Decryption	1.2ms	6.0%	Hardware (AES-NI)
3. Symbolic Check	0.1ms	0.5%	Scalar check
4. **GCN Inference**	**9.0ms**	**45.0%**	**AVX2 (Peak RAM)**
5. Signing (ECDSA)	3.3ms	16.5%	Batch hash sign
Total (Avg)	**20.0ms**	**100%**	End-to-end

Throughput and Memory Optimization: To sustain **10,000 TPS** with $B = 64$, the system must achieve a processing interval of **6.4ms** ($64/10,000s$). We address the dual constraints of compute and memory as follows:

- **Serialized Neural Execution:** Since a single GCN inference (Batch Size 64, Window 50) consumes **118MB**, which is near the 128MB EPC limit, we ensure only **one** neural inference task runs at any physical moment. This prevents memory overflow (EPC swapping) that would occur if multiple batches were processed in parallel.

- **Pipelined Task Overlap:** We utilize the 4 logical threads of the Zen 3 architecture to pipeline non-bottleneck stages. While the GCN engine is occupied by Batch N, the asynchronous threads handle the Decryption and Accumulation of Batch $N+1$ (which consume $<$5MB RAM).
- **Intra-batch Parallelism:** The 9.0ms GCN inference is accelerated by AVX2 SIMD instructions across the assigned vCPUs. The overlapping of I/O and crypto tasks allows the system to meet the 6.4ms throughput cadence in a steady state.

Hyperparameter Sensitivity Analysis

We analyzed the trade-off between detection recall and resource consumption by varying the **History Window Size (W)**.

Table 3: Impact of Window Size (W) on System Performance (Fixed $B = 64$)

Window (W)	**Recall** (%)	**Latency** (ms)	**Peak EPC RAM** (MB)
10	68.4%	12.5	45
30	89.1%	16.2	72
50 (Opt.)	**99.0%**	**20.0**	**118**
100	99.2%	42.5	**$>$128 (Swap)**

Observation: $W = 50$ is selected as the optimal configuration. It provides near-perfect recall (99.0%) while ensuring the **Peak Memory** (118MB) stays within the hardware limits. Extending W to 100 causes the working set to exceed 128MB, triggering catastrophic EPC paging (swapping) and doubling the latency to 42.5ms.

D Dataset and Business Context

This section provides additional details on the industrial significance of the dataset used in our study and the technical pipeline for data transformation and attack injection.

Industrial Scenario: The "Regional Power-to-Economy" Index

In the context of the modern data factor market, granular energy consumption data is a high-value asset. We focus on a real-world industrial scenario:

- **Data Asset:** Raw electricity load data (kilowatts) from 10 distinct industrial plants (e.g., steel, chemical, and manufacturing).
- **Data Product:** A **Regional Industrial Activity Index (RIAI)** (Y_{out}), calculated as a weighted moving average of power loads, serving as a real-time proxy for regional GDP.
- **The Conflict:** While the aggregate index (Y_{out}) is tradable, individual factory loads (D_{raw}) are proprietary. Competitors can use leaked D_{raw} to infer a factory's production schedule and strategic capacity.

Dataset Specifications

We utilize the **Energy-Twin Simulation Dataset (ETSD)**, a high-fidelity synthetic dataset generated by an industrial digital twin platform.

- **Scale:** 10,000 sequential records (approx. 7 days of data at 1-minute granularity).
- **Entities:** 10 factories ($\text{Fac}_1 \ldots \text{Fac}_{10}$) with varying seasonal and diurnal patterns.
- **Features:** Each record contains voltage, current, power factor, and active power.

Data Schema

Table 4 defines the attributes ingested by the VeriAudit Enclave.

Table 4: Data Schema for ETSD Dataset

Attribute	Type	Description
`Timestamp`	Int64	Unix epoch in ms.
`Factory_ID`	String	Anonymized factory identifier.
`V_phase_[A,B,C]`	Float	Three-phase voltage (V).
`I_phase_[A,B,C]`	Float	Three-phase current (A).
`Active_Power`	Float	**Sensitive input feature (kW).**
`RIAI_Index`	Float	The derived data product (Y_{out}).
`Audit_Proof`	String	TEE-generated signature (σ_{PoA}).

Graph Construction and Preprocessing

To enable semantic-aware auditing, raw streams are transformed into a **Dynamic Derivation Graph (DDG)**:

1. **Z-Score Normalization:** All power readings are normalized using moving statistics to eliminate scale variance across different factory sizes.
2. **Temporal Windowing ($W = 50$):** For each time-step t, a window of 50 historical steps is used to capture temporal correlations, forming the feature vector for each node.
3. **Graph Topology:** We construct a star-topology graph where 10 factory nodes are connected to a central "Product Node" representing Y_{out}. Edge weights represent the derivation coefficients.
4. **Batch Formation ($B = 64$):** 64 overlapping windowed graphs are packed into a single tensor for TEE-optimized parallel inference.

Attack Injection Methodology

To evaluate the 99% recall rate reported in Section 5, we simulate **Adaptive Inference Attacks** (Semantic Backdoors).

- **Target:** Leak the load pattern of Fac_1 through the public Y_{out}.
- **Method:** We subtly manipulate the derivation weights $\mathbf{w}$ over time such that the mutual information $\mathcal{I}(Y_{out}; \text{Fac}_1)$ increases.
- **The "Semantic Gap":** These manipulated outputs remain within the "normal" numerical range (passing all symbolic checks) but exhibit an abnormal Pearson correlation ($r > 0.95$) with the sensitive source.

Our GNN is trained specifically to distinguish these malicious correlations from benign aggregate trends.

E Case Study: Forensic Log Analysis

This section provides a detailed breakdown of a forensic log generated by the VeriAudit Enclave. We focus on the specific evasion attempt at time-step $t = 385$ discussed in Section 5.5, illustrating the decision-making process of the Neuro-Symbolic Agent.

Audit Log Structure

In the VeriAudit framework, every transaction is recorded in a cryptographically signed JSON format before the final output is released. The signed log preserves the raw audit trace of this **SOFT_BLOCK** event, where a semantic leak was detected despite passing all deterministic rules.

Log Interpretation and Forensics

The log provides an "X-ray" view of the audit process:

1. **Symbolic Pass:** The attacker manipulated the index such that the value (7540.2) and its volatility ($+0.02\%$) appeared perfectly normal. The Symbolic Engine, which only checks numeric bounds, yielded a `PASS`.
2. **Neural Red-Flag:** The Neural Engine analyzed the embedding of the current window. It detected that the derived product node and the *Fac_1* node exhibited a **cosine similarity of 0.942**. In a benign aggregate index, this similarity should be low (~ 0.150).
3. **High Risk Score:** The resulting risk score of 0.854 (which approximates the mutual information leakage) exceeded the safe threshold of 0.750.
4. **Non-Repudiation:** The `proof_of_audit` section contains a hardware-protected signature. This ensures that the Data Processor cannot deny that the audit took place or tamper with the audit results for regulatory review.

System Monitoring Dashboard

While raw logs are used for forensics, VeriAudit provides a real-time monitoring interface for regulators. The conceptualized dashboard view illustrates the key metrics tracked:

- **Throughput Meter:** Real-time TPS tracking, maintaining $\sim$10,000 TPS.
- **Leakage Heatmap:** A GNN-based graph visualization showing nodes with abnormal synchronization (Fac_1 glowing red in the $t = 385$ event).
- **Audit Latency Histogram:** A distribution of response times centered at 20ms.

F Artifact Availability and Reproducibility

To support the robust empirical validation and system infrastructure claims of our work, the source code, anonymized datasets, and pre-trained models are made available for review. The repository is structured to guide the reviewer through all key steps of the *VeriAudit* framework.

Anonymous Repository Link

The complete codebase, documentation, and assets can be accessed at:

Anonymous Repository:
`https://anonymous.4open.science/r/temp-project-2026-7E24/`

Code Artifacts and System Verification

The package allows for the full reproduction of all experimental figures and serves as demonstrable evidence for our theoretical claims on TEE-native auditing infrastructure.

1. **Algorithmic Core Reproduction (Figures 4, 5, and 6; Table 2):** The main Python scripts (e.g., `veriaudit_pipeline.py`) are located in the root directory to verify reported Precision, Recall, and system efficacy.
2. **TEE Infrastructure Proofs (Appendices A and B):** Specialized scripts demonstrate industrial deployment feasibility:
   - **Resource Constraints:** `Dockerfile` simulates the **128MB EPC** and **2 vCPU** limits, proving architectural efficiency.
   - **Decoupled Architecture:** `sidecar_interceptor_mock.py` demonstrates asynchronous interception logic.
   - **Secure Policy Update:** `remote_attestation_mock.py` simulates the Sigma protocol for TEE integrity.
   - **Deployment Format:** `export_to_onnx.py` converts models to ONNX for TEE inference runtimes.
3. **Data Anonymity:** `sample_data.csv` contains 10,000-step time series, anonymized by shifting timestamps to **2070** and masking regional codes (`Reg_XX`) per double-blind policy.