Resilient Privacy-Preserving Collaborative Predictive Maintenance through a Federated Multi-Layer Architecture

Jonah Giglio[1], Salvatore D'Antonio[1] and Giovanni Maria Cristiano[1]

Abstract— **The adoption of predictive maintenance in Industry 4.0 environments generates valuable operational data that could benefit from collaborative learning across multiple industrial facilities. However, competitive concerns and regulatory frameworks such as GDPR and NIS2 make centralized data aggregation increasingly impractical and non-compliant. Federated learning enables decentralized collaboration but requires confidentiality guarantees during model aggregation, and transparent governance mechanisms for multi-party trust. This paper proposes a resilient architecture integrating federated learning with trusted execution environments to protect aggregation processes, and blockchain for immutable governance, automatic model versioning, and operational continuity under adversarial or fault conditions. Operating across edge-fog-cloud tiers, the system enables industrial sites to collectively train models without sharing raw data. We present a formal system design with analysis of privacy guarantees, governance mechanisms, and fault-resilience properties — ensuring that neither a compromised sensor nor a corrupted model halts production operations. Analytical evaluation demonstrates that the security stack introduces a 123% latency overhead over standard federated averaging, remaining within acceptable bounds for daily or weekly industrial training cycles, while communication overhead remains below 1% of total traffic.**

Keywords: Federated Learning, Collaborative Maintenance, Trusted Execution Environments, Blockchain, Industry 4.0

I. Introduction

The convergence of Industry 4.0 and advanced manufacturing has generated unprecedented volumes of operational data from industrial equipment [1]. Modern manufacturing facilities are instrumented with extensive sensor networks that continuously monitor equipment health through measurements of vibration, temperature, pressure, energy consumption, and numerous other parameters [2]. These data, when combined with maintenance logs and process parameters, contain rich patterns that can reveal early indicators of equipment degradation and predict failure modes. Machine Learning (ML) models trained on such data can identify subtle anomalies invisible to rule-based systems and optimize maintenance schedules. The potential value multiplies when learning can occur across multiple facilities operating similar equipment [3].

Traditional approaches to industrial ML have relied on centralized data aggregation, where operational data is collected into a single repository for model training. However, this centralized paradigm faces barriers in contemporary industrial environments [4]. Manufacturing companies operate in highly competitive ecosystems where sensor readings and maintenance patterns reveal proprietary information about process efficiency, equipment utilization, and operational practices [5]. Sharing such data—even among plants of the same organization—risks exposing competitive insights, while in supply chain scenarios involving multiple companies, the reluctance to centralize data becomes even more pronounced. Regulatory frameworks compound these challenges. The European Union's GDPR establishes principles of data minimization and purpose limitation, while the NIS2 Directive mandates enhanced cybersecurity measures for critical infrastructure operators with explicit requirements for supply chain security. These frameworks make centralized data aggregation non-compliant, as concentrating sensitive operational data creates both security vulnerabilities and regulatory risks.

Federated Learning (FL) enables multiple industrial sites to collectively train predictive maintenance models without centralizing raw data. Each facility trains a local model copy and shares only model updates, rather than raw data [6]. The distributed nature of FL aligns naturally with cloud computing paradigms, which provide the multi-tier infrastructure necessary for industrial-scale deployment. Edge devices at industrial sites perform data collection and preprocessing; fog computing nodes execute local training while keeping sensitive data within facility boundaries; and cloud infrastructure coordinates aggregation and orchestrates training rounds across distributed participants. However, three critical concerns remain. First, participants need verifiable guarantees that their model updates cannot be inspected during aggregation. Second, a corrupted or poisoned model must never reach deployment — and if detected, the system must automatically revert to the last verified version without interrupting production. Third, multi-party collaborations require transparent governance and verifiable accountability. This paper proposes an architecture that addresses all three concerns simultaneously. Our work makes the following contributions:

- A comprehensive multi-layer architecture that integrates FL with TEEs and blockchain, for secure, collaborative predictive maintenance in competitive environments.
- A systematic framework defining communication protocols, security guarantees, fault-resilience mechanisms, and governance policies to ensure operational continuity under adversarial conditions.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III presents the reference scenario. Section IV defines the threat model. Sections V and VI describe the architecture and performance analysis. Section VII provides a security analysis. Section VIII concludes the paper.

[1]University of Naples "Parthenope", Centro Direzionale, Isola C4, Naples, 80133, Italy. Email: *jonah.giglio@phds.uniparthenope.it*, *salvatore.dantonio@uniparthenope.it*, *giovannimaria.cristiano@phds.uniparthenope.it*

II. Related Work

Research in predictive maintenance for Industrial IoT has primarily focused on three separate technological domains: edge computing architectures, blockchain-based governance, and trusted execution environments. However, no existing work integrates these three pillars to simultaneously address latency, trust, and security requirements.

Edge and fog computing approaches have demonstrated the feasibility of distributed AI inference in industrial settings. Zhou et al. [7] survey edge intelligence architectures, showing how local processing reduces latency but noting the inability of resource-constrained devices to train complex models. Li et al. [8] propose on-demand deep learning acceleration at the edge, yet their work assumes pre-trained models are securely distributed—precisely the trust gap we address. Federated learning frameworks [9], [10] enable collaborative model training across distributed nodes while preserving data locality, but lack mechanisms to verify model integrity or prevent poisoning attacks during aggregation.

Blockchain integration in IIoT has focused primarily on supply chain traceability and data provenance. While early studies identify consensus latency as a barrier [11] or propose IoT security without addressing AI workloads [12], recent blockchain-based federated learning approaches [13], [10] use smart contracts to coordinate distributed training. Yet, these systems execute AI computations outside the blockchain, creating a trust gap where model training cannot be cryptographically verified [14]. Furthermore, public blockchain consensus mechanisms introduce latencies incompatible with industrial control loops, while permissioned solutions lack integration with secure execution environments.

Trusted execution environments have emerged as a solution for confidential computing in cloud environments. Intel SGX [15] provides hardware-isolated enclaves with remote attestation enabling cryptographic verification of enclave integrity [16]. While works like Brenner et al. [17] and Tramer and Boneh [18] apply TEEs to secure coordination and ML inference, they focus on isolated cloud deployments. Critically, they do not address the end-to-end trust chain for multi-plant systems and lack mechanisms for auditable model deployment to edge devices [19].

As illustrated in Table I, our architecture uniquely bridges these gaps. By combining edge-fog distribution for performance, Intel TDX as a cryptographic trust anchor for training, and Hyperledger Fabric for immutable governance, we establish an unbroken chain of trust. To our knowledge, this is the first framework to provide a cryptographically verifiable pipeline from sensor data acquisition to auditable multi-plant model deployment.

TABLE I
Comparison with State-of-the-Art Approaches

Approach	Edge	TEE	Blockchain	E2E Security
Edge Intelligence [7]	✓	×	×	Low
Federated Learning [10]	✓	×	×	Medium
Blockchain-FL [13]	×	×	✓	Medium
TEE-ML [18]	×	✓	×	Medium
Our Work	✓	✓	✓	**High**

III. Reference Scenario

To ground the threat model and architectural design in a concrete operational context, we consider a representative industrial deployment: four manufacturing companies collaborating on predictive maintenance for CNC machining centers within an automotive supply chain. Each facility operates independent edge gateways collecting vibration, temperature, motor current, and acoustic emission data. Companies are willing to improve their maintenance models through collaborative learning, but cannot share raw sensor readings due to competitive concerns and regulatory constraints under GDPR and NIS2. In this setting, fog servers deployed on-premise at each plant perform local model training on plant-specific datasets. Only weight updates are transmitted to a shared cloud aggregation service, where a global model is refined using Federated Learning (FL). The resulting model is distributed back to all participants via a blockchain-governed pipeline, with automatic rollback to the previous approved version if the new model fails consortium validation thresholds. At one facility, sensors detect elevated vibration on a CNC spindle. The collaborative LSTM model—trained on patterns from all four plants—predicts spindle bearing failure within 72 hours, triggering an automatic maintenance workflow. Proactive replacement reveals early-stage raceway spalling, preventing unplanned shutdown. The entire intervention chain—sensor readings, prediction, actions, and outcome—is recorded on-chain with immutable timestamps and becomes labeled data for the next training round. All consortium members benefit without exposing proprietary operational data.

IV. Threat Model

The collaborative scenario introduce a broad attack surface spanning physical infrastructure, network communications, software components, and human actors. Unlike single-operator deployments, no single entity can be assumed fully trusted, and the system must remain secure even when a subset of participants behaves maliciously. A rigorous threat model is therefore essential to define trust boundaries and security requirements before describing the architecture.

Adversary Model. We consider a realistic threat model for industrial environments. At the edge layer, an adversary may gain physical access to edge gateways to extract cryptographic keys, manipulate firmware, or install malicious sensors. Software-level attacks may inject crafted feature vectors or intercept TPM (Trusted Platform Module) attestation flows. Malicious participants may corrupt training through data poisoning: label flipping on maintenance logs, feature manipulation to bias learning, or adversarial examples targeting specific prediction errors. Byzantine fog nodes may submit malicious weight updates via gradient inversion, model replacement, or convergence sabotage. At the cloud layer, attackers may exploit hypervisor vulnerabilities or gain privileged access to inspect model parameters or cached data. Malicious insiders with legitimate credentials may attempt unauthorized model deployment, exfiltrate proprietary data, or modify smart contract

logic. Network-level threats include man-in-the-middle attacks on TLS channels, replay attacks resubmitting stale attestation quotes or weight updates, and denial-of-service targeting blockchain consensus or IPFS (InterPlanetary File System) availability.

Trust Assumptions. We assume standard cryptographic primitives—AES-256-GCM, ECDSA P-256, SHA-256—are computationally secure. TPM and TDX (Trusted Domain Extensions) hardware trusted computing bases remain uncompromised. Sophisticated physical tampering or microarchitectural side-channel attacks are explicitly out of scope; a hardware breach at this level would compromise our confidentiality guarantees. The blockchain maintains an honest majority with fewer than $f < n/3$ Byzantine consensus nodes. Finally, regarding AI safeness, defending against highly stealthy, coordinated backdoor attacks remains an open challenge in federated learning. Certificate authorities and Intel's attestation verification service are trusted. ZKP (Zero-Knowledge Proof) systems provide computational soundness. Under these assumptions, the system guarantees operational continuity: any detected violation triggers component isolation and automatic fallback, rather than full pipeline termination.

V. Proposed Architecture

The architecture spans four layers with distinct roles, illustrated in Fig. 1. Edge gateways (TPM-equipped) authenticate sensors, preprocess raw data, and run local inference — raw data never leaves the plant. Fog servers, deployed on-premise at each facility, perform local model training and generate ZKP-verified weight updates. Cloud enclaves (Intel TDX) aggregate weight updates from all participants inside a hardware-isolated environment, acting as the trust anchor for the entire pipeline. The blockchain (Hyperledger Fabric) provides governance and auditability: smart contracts verify attestation evidence, enforce multi-party endorsement, and emit immutable events that trigger model distribution.

A. Architectural Overview

Before operational deployment, each participating organization (Company #1, #2, ..., #n) registers with the Consortium Governance smart contract on the blockchain (⓪), obtaining cryptographic credentials, role assignments, and participation permissions required for federated learning.

Edge-Fog Pipeline: Data Collection and Local Training. Sensors continuously collect multivariate time-series data $\mathbf{x}_t \in \mathbb{R}^d$ at frequencies from 1 Hz to 10 kHz and transmit measurements to the local gateway with attached X.509 certificates (①). The gateway, equipped with TPM 2.0, receives sensor data and verifies sensor authenticity by querying the Local Trust Store, a local database containing authorized sensor X.509 certificates (②). Unregistered sensors are rejected and blacklisted in the Local Trust Store; inference continues uninterrupted using the locally cached model $\mathscr{M}_{\text{edge}^r}$, ensuring operational continuity. Authenticated data undergoes preprocessing through a feature extraction function $\phi : \mathbb{R}^{d\times n} \to \mathbb{R}^{d'}$ that computes:

$$\mathbf{f}_i = \phi(\mathbf{X}_i) = [\text{RMS}(\mathbf{X}_i), \text{FFT}(\mathbf{X}_i), \text{kurt}(\mathbf{X}_i), \ldots] \quad (1)$$

where $d' \ll d \cdot n$ achieves dimensionality reduction while preserving diagnostic information. Simultaneously, the gateway executes local inference using a lightweight cached model $\mathscr{M}_{\text{edge}}$ that flags anomalies when the reconstruction error exceeds threshold τ:

$$\text{alert} = \mathbb{1}\left[\|\mathbf{f}_i - \mathscr{M}_{\text{edge}}(\mathbf{f}_i)\|_2 > \tau\right] \quad (2)$$

Critical anomalies trigger alert generation, notifying Human-in-the-Loop (HIL) operators for immediate intervention (③–⑤). The gateway then initiates an mTLS 1.3 handshake with the on-premise fog server (⑥). The fog's Security & Attestation Gate validates the gateway's TPM attestation quote against the PCR Database (Platform Configuration Registers) to verify firmware and OS integrity, and authenticates X.509 certificates via the IAM Database (⑦). Upon successful authentication, the gateway transmits preprocessed features $\mathbf{f}_i$—not raw sensor data $\mathbf{X}_i$—to the fog layer, preserving data sovereignty (⑧). Received features are processed by the Byzantine Fault Detector, which communicates with the IAM Database to identify and exclude potentially compromised gateways based on attestation history and behavioral patterns (⑨). Validated data enters the local training module, where the fog performs Stochastic Gradient Descent on the global model $\mathscr{M}_g^{(r)}$ using plant-specific dataset $\mathscr{D}_p = \{(\mathbf{f}_1, y_1), \ldots, (\mathbf{f}_k, y_k)\}$ combining features with maintenance labels (⑩):

$$\mathscr{M}_p^{(r+1)} = \mathscr{M}_g^{(r)} - \eta \nabla_\theta \mathscr{L}(\mathscr{M}_g^{(r)}, \mathscr{D}_p) \quad (3)$$

where η is the learning rate and $\mathscr{L}$ is the cross-entropy loss for classification or MSE for regression. The fog extracts weight updates:

$$\Delta\mathbf{w}_p = \mathbf{w}_p^{(r+1)} - \mathbf{w}_g^{(r)} \quad (4)$$

and generates a Zero-Knowledge Proof (ZKP) π_p cryptographically attesting that $\Delta\mathbf{w}_p$ was computed correctly from verified data without revealing proprietary dataset $\mathscr{D}_p$ (⑪). The fog initiates authentication with the cloud infrastructure via mTLS and mutual attestation, then transmits the tuple $(\Delta\mathbf{w}_p, \pi_p, \text{TPM}_{\text{quote}})$ to the cloud's Security & Attestation Gate (⑫, ⑬). Concurrently, all other participating organizations (Companies #2, #3, ..., #n) execute identical steps and submit their weight updates to the cloud (⑭).

Cloud Aggregation: Secure Federated Learning. Operating within Intel TDX-protected enclaves, the cloud executes ZKP verification for all proofs $\{\pi_1, \pi_2, \ldots, \pi_P\}$ and performs Federated Averaging through the Global Model Aggregator & Training module (⑮):

$$\mathbf{w}_g^{(r+1)} = \sum_{p=1}^{P} \frac{|\mathscr{D}_p|}{|\mathscr{D}|} \mathbf{w}_p^{(r+1)} \quad (5)$$

where P is the number of participating plants, $|\mathscr{D}_p|$ is the plant-specific dataset size, and $|\mathscr{D}| = \sum_{p=1}^{P} |\mathscr{D}_p|$ is the total dataset size. Aggregation occurs in encrypted memory inaccessible to the hypervisor. The resulting model $\mathscr{M}_g^{(r+1)}$ is sealed using TDX-protected keys (⑯) and cryptographically signed by the Model Signer. The model is uploaded to IPFS, generating a Content Identifier (CID). The cloud submits

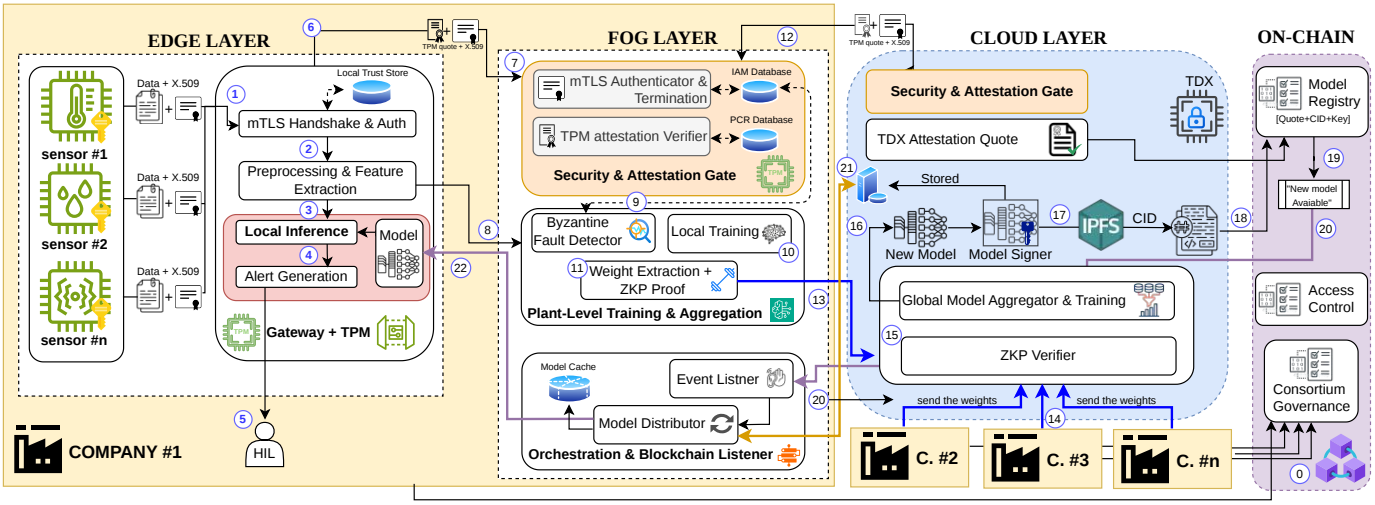

Fig. 1. Overview of the proposed Edge-Fog-Cloud hierarchical pipeline for secure federated learning and blockchain-based model management.

to blockchain: the CID, a TDX Attestation Quote $\mathscr{A}_{\text{TDX}}$ embedding MRENCLAVE, MRSIGNER, and $H(\mathscr{M}_g^{(r+1)})$, along with performance metadata (⑱).

Blockchain Governance and Model Distribution. The Model Registry smart contract verifies: (1) TDX attestation validity (Intel signature verification, MRENCLAVE correctness), (2) model accuracy exceeds consortium-defined thresholds, and (3) collects multi-organization endorsements (m-of-n cryptographic signatures). Upon successful verification, the model is registered with status 'Approved' and replaces the active version; if verification fails, the previous approved model remains active and a new training round is automatically scheduled — production is never interrupted. Then, the smart contract emits a "New Model Available" blockchain event containing the CID (⑲, ⑳). The fog's Event Listener within the Orchestration & Blockchain Listener module receives the blockchain event (㉑). The Model Distributor retrieves the CID from the event, downloads the model from IPFS, verifies the model hash against the blockchain-registered hash, and distributes the verified model to edge gateways for updated local inference (㉒). The Access Control and Consortium Governance contracts enforce role-based permissions and democratic decision-making throughout the pipeline, establishing a cryptographically verifiable chain of custody — from sensor acquisition to verified deployment — secured by hardware-rooted trust anchors (TPM at edge, TDX at cloud) and immutable blockchain audit trails.

B. Blockchain Layer

Hyperledger Fabric operates as a permissioned network with Raft consensus ordering, achieving 200–500 ms finality. Three smart contracts implement the governance pipeline.

Model Registry. The *ModelRegistry* stores structured records for each trained model (Table II). Registration requires: (1) valid TDX attestation matching MRENCLAVE, (2) accuracy exceeding consortium thresholds, and (3) m-of-n multi-organization endorsement. Approved models are retrievable by CID for fog distribution.

TABLE II
STRUCTURE OF THE *ModelRecord* SMART CONTRACT ENTITY

Field	Type	Description
`modelHash`	`bytes32`	SHA-256 of weights
`ipfsCID`	`string`	IPFS identifier
`tdxAttestation`	`bytes`	TDX quote
`accuracy`	`uint256`	Validation accuracy
`endorsers`	`address[]`	Approving orgs
`status`	`enum`	Model state

Access Control. The *AccessControl* contract enforces role-based permissions: plant operators submit training data and trigger maintenance; ML engineers initiate training rounds; auditors have read-only access. Critical operations require m-of-n multi-signature approval.

Maintenance Orchestrator. When edge gateways detect high-confidence failure predictions, they invoke *triggerMaintenance(plantID, prediction, confidence)*, creating immutable event records consumed by external CMMS systems. Post-maintenance results submitted via training rounds.

VI. PERFORMANCE ANALYSIS

We evaluate the proposed architecture through analytical simulation, comparing it against a **baseline** defined as standard Federated Averaging [9] without TEE protection or blockchain governance. This baseline serves as a practical lower bound to isolate and estimate the overhead introduced by each security layer: TPM attestation at the edge, ZKP generation at the fog, TDX enclave isolation at the cloud, and Hyperledger Fabric consensus. The goal of this comparison is to evaluate the general trade-offs of our security mechanisms, rather than claiming a performance advantage over other protected architectures. Three metrics are evaluated: end-to-end latency (i) per training round, communication bandwidth (ii) per participant, and computational overhead (iii) per layer.

A. Simulation Configuration

The analytical model assumes 4 participating plants, each operating 10 edge gateways with TPM 2.0 modules processing 5

sensors at 1 kHz. Fog servers deploy on-premise (16 GB RAM, Intel Xeon E5), representing commodity hardware accessible to manufacturing facilities without specialized infrastructure investment. Cloud infrastructure uses TDX-enabled VMs (Intel 4th Gen Xeon Scalable), the current generation supporting confidential computing workloads. The federated learning model comprises 3.5M parameters (LSTM rchitecture), trained on 1,000 samples per plant — a conservative estimate consistent with annual maintenance log volumes in automotive supply chains. Network connectivity assumes 1 Gbps Ethernet. Security latencies are derived from published benchmarks: TPM attestation (200 ms) [16], ZKP generation (100 ms/MB), TDX overhead (+32%) [18], Hyperledger Fabric Raft consensus (350 ms).

B. Latency Analysis

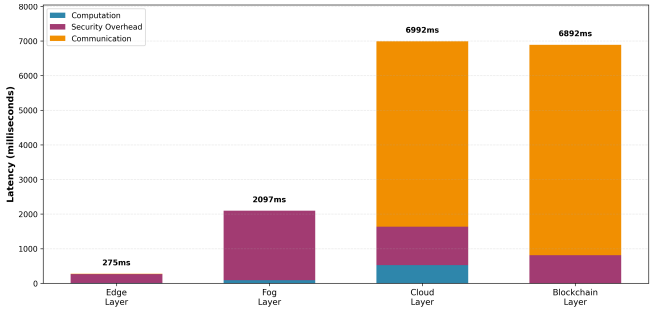

Fig. 2. End-to-end latency breakdown across the edge, fog, cloud, and blockchain layers (including security overheads)

Fig. 2 presents the end-to-end latency breakdown for a complete federated learning round. The total round duration is approximately 16.3 s: Edge layer operations contribute 275 ms (1.7%), with the majority spent on mTLS handshake and TPM attestation. Fog layer processing requires 2.1 s (12.9%) for local training, Byzantine fault detection, and ZKP proof generation — the latter accounting for 1.35 s due to cryptographic proof construction over 13.4 MB weight updates. Cloud aggregation within TDX introduces 7.0 s (43.0%), primarily attributed to IPFS model upload (5.4 s) and secure aggregation overhead (0.5 s TDX isolation plus 1.0 s ZKP verification across 4 participants). Blockchain operations require 6.9 s (42.4%), split between smart contract verification (0.8 s) and model distribution (6.0 s). Compared to the baseline (7.3 s), security mechanisms introduce 9.0 s of additional overhead, representing a 123% increase. This remains acceptable for daily or weekly industrial training cycles, where round duration is not time-critical. Edge-level anomaly detection maintains $<$10 ms latency via cached models, preserving real-time operational safety independently of training round frequency.

C. Communication Overhead

Fig. 3 evaluates network bandwidth requirements as a function of model size. For the baseline 3.5M-parameter model (13.4 MB), total communication overhead reaches 80.2 MB per training round: fog-to-cloud weights (53.4 MB), cloud-to-IPFS storage (13.4 MB), and fog-to-edge distribution (13.4 MB).

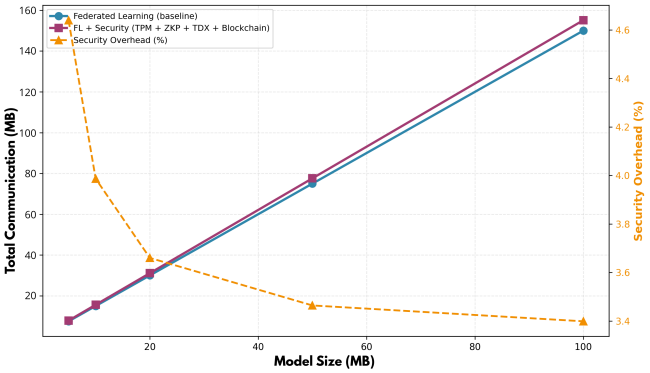

Fig. 3. Communication overhead versus model size for baseline FL and the proposed secure flow

Security overhead adds only 0.16 MB (0.2%): X.509 certificates, TPM quotes, and ZKP proofs. Scalability analysis shows linear growth with model size: scaling from 5 MB to 100 MB increases total traffic from 40 MB to 615 MB, maintaining a consistent 6$\times$ multiplier due to multi-participant distribution. Security overhead remains nearly constant at 0.15–0.20 MB regardless of model size, representing decreasing relative impact ($<$1% for models above 100 MB).

D. Computational Complexity

Fog training dominates at 0.14 TFLOPS (99.9% of total), cloud aggregation requires 42 GFLOPS, and edge operations contribute $<$1 GFLOPS. Security operations (ZKP, TPM, TDX) introduce cryptographic overhead but negligible floating-point computation relative to model training. The architecture achieves comprehensive security guarantees at a 123% latency overhead versus the unsecured baseline — a trade-off justified by the threat model requirements and regulatory compliance obligations (GDPR, NIS2).

VII. Security Analysis

We analyze security properties with respect to the threat model defined in Section IV, demonstrating that the architecture satisfies the four fundamental security requirements. **Confidentiality.** Raw sensor data remains within facilities, processed locally at edge and fog layers — only weight updates are transmitted for aggregation. Intel TDX provides hardware-enforced memory isolation during cloud aggregation, preventing hypervisor or administrator access to model parameters even in untrusted cloud infrastructure. TDX remote attestation enables cryptographic verification that aggregation executed on genuine hardware running approved code. This design enforces data minimization as required by GDPR Article 25, ensuring that no raw operational data crosses organizational boundaries at any stage of the pipeline.

Integrity. Model hashes, TDX attestation evidence, and training metadata are immutably recorded on blockchain, ensuring only cryptographically verified models can be deployed. Smart contracts enforce role-based access control, restricting approval and deployment permissions to authorized roles. Data poisoning and model tampering are constrained by secure TEE aggregation combined with ZKP-verified weight updates and

Byzantine fault detection at the fog layer, enabling detection and accountability for malicious participants.

Availability. Edge devices retain locally cached models, enabling autonomous operation during cloud or blockchain outages. Combined with automatic model rollback via blockchain versioning, this guarantees operational continuity at the plant level regardless of whether the failure originates from network disruption, a poisoned model, or a compromised participant. The permissioned blockchain employs fault-tolerant Raft consensus, supporting graceful degradation under partial node failures. Blockchain-based model versioning prevents deployment of stale or malicious models, maintaining service integrity across training rounds.

Auditability. All critical events - model registration, training rounds, attestation results, and deployment decisions — are permanently logged on blockchain with cryptographic timestamps. This immutable audit trail enables forensic analysis and compliance verification with GDPR Article 25 and the NIS2 Directive, establishing a verifiable chain of custody from sensor acquisition through collaborative training to predictive maintenance actions at the plant level.

VIII. Conclusion

This paper presented a multi-layer architecture for privacy-preserving collaborative predictive maintenance in Industry 4.0 environments. By integrating FL with TEEs and blockchain-based governance, the proposed framework enables multiple industrial stakeholders to collaboratively train machine learning models without sharing raw operational data or sacrificing data sovereignty. The architecture leverages edge and fog computing to preserve locality and low latency, while confidential computing in the cloud ensures secure model aggregation and training even in untrusted infrastructures. The use of blockchain provides immutable auditability, transparent governance, and verifiable model provenance, addressing key trust challenges in multi-party industrial collaborations. The security analysis demonstrates that the system satisfies essential requirements in terms of confidentiality, integrity, availability, and accountability under a realistic adversary model. Overall, this work contributes a practical and secure foundation for deploying collaborative AI in competitive industrial ecosystems. The performance analysis demonstrates a 123% latency overhead over the unsecured baseline, which remains acceptable for daily or weekly training cycles. However, the current evaluation relies on analytical simulation; future work will prioritize empirical validation on real industrial deployments and investigate scalability under heterogeneous workload conditions.

Acknowledgment

This work has been partially funded by the research projects “Approccio User-friendly integrato per Diagnosi, Assistenza e Cura Efficaci” - AUDACE (CUP B69J23006050007), MIRABILE (CUP C67B23000300004) - CDS001108 “Filiera Automotive 4.0” – PNRR M1C2 - 5.2 Filiere produttive - “Automotive” NextGenerationEU and ISTINTO (CUP C85I23000400004) - CDS001133 “Steel Network 4.0” – PNRR M1C2 - 5.2 Filiere produttive – “Metallo ed elettromeccanica” NextGenerationEU.

References

[1] R. Rai, M. K. Tiwari, D. Ivanov, and A. Dolgui, “Machine learning in manufacturing and industry 4.0 applications,” pp. 4773–4778, 2021.

[2] M. Ghobakhloo, “Industry 4.0, digitization, and opportunities for sustainability,” *Journal of cleaner production*, vol. 252, p. 119869, 2020.

[3] A. Angelopoulos, E. T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, and T. Zahariadis, “Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects,” *Sensors*, vol. 20, no. 1, p. 109, 2019.

[4] A. Sircar, K. Yadav, K. Rayavarapu, N. Bist, and H. Oza, “Application of machine learning and artificial intelligence in oil and gas industry,” *Petroleum Research*, vol. 6, no. 4, pp. 379–391, 2021.

[5] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, and J. Barbosa, “Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges,” *Computers in industry*, vol. 123, p. 103298, 2020.

[6] G. M. Cristiano, S. D’Antonio, and F. Uccello, “A novel approach for securing federated learning: Detection and defense against model poisoning attacks,” in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2024, pp. 664–669.

[7] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[8] E. Li, L. Zeng, Z. Zhou, and X. Chen, “Edge AI: On-demand accelerating deep neural network inference via edge computing,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, 2020.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2017, pp. 1273–1282.

[10] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 12:1–12:19, 2019.

[11] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with iot. challenges and opportunities,” *Future Generation Computer Systems*, vol. 88, pp. 173–190, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17329205

[12] N. M. Kumar and P. K. Mallick, “Blockchain technology for security issues and challenges in IoT,” *Procedia Computer Science*, vol. 132, pp. 1815–1823, 2018.

[13] H. Kim, J. Park, M. Bennis, and S. Kim, “Blockchained on-device federated learning,” *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020, publisher Copyright: © 1997-2012 IEEE.

[14] L. Coppolino, G. M. Cristiano, S. D’Antonio, J. Giglio, G. Mazzeo, and L. Romano, “A blockchain solution for decentralized content verification and its application to deepfake detection and fintech credit scoring,” *Blockchain: Research and Applications*, p. 100406, 2025.

[15] V. Costan and S. Devadas, “Intel sgx explained,” Cryptology ePrint Archive, Report 2016/086, 2016, available: https://eprint.iacr.org/2016/086.

[16] S. Johnson, V. Scarlata, C. Rozas, E. Brickell, and F. Mckeen, “Intel Software Guard Extensions: EPID provisioning and attestation services,” Intel Corporation, White Paper, 2016.

[17] S. Brenner, C. Wulf, D. Goltzsche, N. Weichbrodt, M. Lorenz, C. Fetzer, P. Pietzuch, and R. Kapitza, “SecureKeeper: Confidential ZooKeeper using Intel SGX,” in *Proc. ACM/IFIP/USENIX Middleware*, 2016, pp. 14:1–14:13.

[18] F. Tramèr and D. Boneh, “Slalom: Fast, verifiable and private execution of neural networks in trusted hardware,” 2019. [Online]. Available: https://arxiv.org/abs/1806.03287

[19] G. M. Cristiano, S. D’Antonio, J. Giglio, G. Mazzeo, and L. Romano, “Enhancing the security of rollup sequencers using decentrally attested tees,” *IEEE Transactions on Emerging Topics in Computing*, 2026.