ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

A Decentralized Artificial Intelligence with Confidential Computing and Smart Contract-Based Trust Verification

D. A. Onuma[1], I. M. Ossai[2], A. H. Mini[3], B. M. Kpobe[4]

[1, 2, 3, 4] Department of Computer Science

[1, 2, 3] Rivers State University, Nkpolu-Oroworukwo, Port Harcourt, Rivers State, Nigeria.

DOI: https://doi.org/10.5281/zenodo.24105049

Published Date: 10-November-2026

Abstract*: Decentralized artificial intelligence enables collaborative learning without centralized data collection but remains vulnerable to model poisoning, malicious participants, and insecure execution environments. This study developed a Decentralized Artificial Intelligence Framework with Confidential Computing and Smart Contract-Based Trust Verification to strengthen privacy, integrity, and scalability. Design Science Research Methodology (DSRM) guided framework development and evaluation, while Object-Oriented Analysis and Design Methodology (OOADM) supported modular system design. The framework integrated peer-to-peer learning, Trusted Execution Environments, remote attestation, blockchain-anchored model updates, smart contracts, and adaptive reputation management. Experimental evaluation used seven benchmark datasets: CICIDS2017, CSE-CIC-IDS2018, UNSW-NB15, CICIoT2023, TON_IoT, Bot-IoT, and NSL-KDD. Assessment covered classification effectiveness, trust verification, privacy, attack resilience, computational overhead, latency, throughput, and scalability. Accuracy and F1-score generally exceeded 97%, while integrity verification exceeded 99%. Compared with the strongest evaluated baseline, blockchain throughput increased by 67.1% and maximum supported nodes by 60.0%, while confidential-execution overhead, false-trust rate, and data-leakage rate decreased by 54.4%, 52.3%, and 51.7%, respectively. Detection of model-poisoning, Byzantine, Sybil, and malicious-node attacks also improved, although residual attack success remained. These findings demonstrate the potential of integrating confidential execution and programmable trust governance to secure collaborative AI in distributed cloud and IoT environments. Further operational evaluation is required to establish real-world effectiveness.

***Keywords*: Decentralized artificial intelligence; confidential computing; smart contracts; trust verification; DSRM; OOADM.**

1. BACKGROUND OF THE STUDY

Artificial Intelligence (AI) is transforming healthcare, finance, manufacturing, transportation, cybersecurity, and smart cities through intelligent, data-driven decision-making. However, the growth of cloud computing, edge computing, the Internet of Things (IoT), and cyber-physical systems has increased data volume and distribution, exposing the limitations of centralized machine learning. Centralized architectures create privacy risks, communication bottlenecks, regulatory challenges, and single points of failure [42, 61].

Decentralized Artificial Intelligence (DAI) addresses these limitations by enabling geographically distributed participants to collaboratively train models without transferring raw data to a central repository. This approach improves privacy, scalability, communication efficiency, and resilience, making it suitable for sensitive applications such as healthcare, finance, industrial IoT, transportation, and cybersecurity [35, 51, 57].

Despite these benefits, decentralized learning introduces security and trust challenges because participants exchange model updates directly through peer-to-peer networks. Malicious nodes can manipulate updates, launch model poisoning, Byzantine, and Sybil attacks, or compromise collaborative learning integrity [31, 61, 5, 9, 20]. Lalitha [35] demonstrated fully decentralized federated learning through gossip-based communication, eliminating the centralized parameter server. However, the framework lacked mechanisms for verifying participant trustworthiness and model-update integrity, as well as protection against runtime attacks, compromised operating systems, and privileged access.

Confidential Computing (CC) offers complementary protection through Trusted Execution Environments (TEEs), including Intel SGX, Intel TDX, AMD SEV-SNP, and ARM CCA. Unlike conventional encryption, which protects data at rest and in transit, confidential computing protects sensitive information during execution, mitigating attacks from compromised operating systems, hypervisors, cloud administrators, and privileged insiders [17, 23, 24, 16, 43]. Nevertheless, its integration into decentralized AI architectures remains limited.

Similarly, **blockchain technology** provides decentralized trust through distributed ledgers and smart contracts, enabling participant authentication, access control, model provenance, incentive management, and immutable auditing [1, 4, 40, 51, 62]. Although blockchain improves transparency and accountability, it cannot independently protect AI computations outside its consensus layer. Consequently, model training and inference may remain vulnerable to runtime attacks.

Recent research has therefore highlighted the potential of integrating decentralized AI, confidential computing, and blockchain to establish trustworthy collaborative intelligence [61, 56, 24, 63]. However, existing solutions remain fragmented: some combine decentralized learning with blockchain but omit confidential execution, while others integrate confidential computing with blockchain without addressing collaborative AI. Furthermore, existing frameworks provide limited support for continuous trust assessment, adaptive reputation management, model-integrity verification, autonomous security enforcement, and interoperability across heterogeneous TEEs [16, 17, 23, 40, 43, 51, 57, 61].

These limitations reveal a research gap: the absence of a comprehensive architecture integrating decentralized collaborative learning, hardware-enforced confidential execution, and blockchain-based smart contract trust verification to provide end-to-end security, privacy, transparency, scalability, and trustworthy governance. This gap constrains the deployment of decentralized AI in sensitive environments, including healthcare, finance, industrial automation, critical infrastructure, and cyber defence [31, 56, 61, 63].

To address this gap, this study develops a **Decentralized Artificial Intelligence Framework with Confidential Computing and Smart Contract-Based Trust Verification** for secure, privacy-preserving, and trustworthy collaborative machine learning. Specifically, it aims to:

1. Design a peer-to-peer decentralized learning architecture that eliminates centralized coordination.
2. Integrate Trusted Execution Environments to protect AI training and inference during execution.
3. Develop blockchain-enabled smart contracts for participant authentication, model-integrity verification, access control, incentive management, and immutable auditing.
4. Implement adaptive trust verification to detect malicious participants, validate model updates, and mitigate collaborative learning attacks.
5. Evaluate the framework using learning accuracy, privacy preservation, trust verification, attack resilience, computational overhead, scalability, latency, throughput, and system reliability.

By integrating decentralized AI, confidential computing, and smart contract-based trust verification, the proposed framework seeks to overcome existing security limitations and provide a scalable, trustworthy collaborative AI architecture for next-generation distributed intelligent applications [24, 40, 56, 61, 63].

2. LITERATURE REVIEW

2.1 Conceptual Review

2.1.1 Artificial Intelligence

Artificial Intelligence (AI) is a branch of computer science concerned with developing systems capable of performing tasks that typically require human intelligence, including learning, reasoning, problem-solving, perception and decision-making. AI uses computational models and algorithms to identify patterns, analyse information and make autonomous decisions across domains such as healthcare, finance, cybersecurity, manufacturing and transportation [52, 28].

AI has evolved from early symbolic reasoning and rule-based expert systems into data-driven machine learning and advanced deep learning. Machine Learning (ML) enables systems to learn patterns from data through supervised, unsupervised, semi-supervised and reinforcement learning approaches [8, 44]. Deep Learning (DL), a subset of ML, employs multilayer neural networks such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) and Transformers to model complex relationships and extract features automatically [28, 36].

The development of distributed computing has further introduced Distributed Artificial Intelligence (DAI), in which multiple autonomous computing entities collaborate through distributed learning, reasoning and decision-making. Unlike centralized AI, DAI distributes computational activities across cloud servers, edge devices, IoT networks and peer-to-peer systems, improving scalability, fault tolerance and data privacy [35, 61]. Collaborative AI similarly enables organizations to share learned knowledge without necessarily exchanging their raw datasets [51, 57].

The AI lifecycle includes data collection, preprocessing, feature engineering, model selection, training, validation, deployment, monitoring and continuous improvement. Effective governance throughout these stages is necessary to maintain reliability, fairness, explainability, security and regulatory compliance [22, 48, 52].

Despite its benefits, AI remains vulnerable to adversarial attacks, data and model poisoning, membership inference, model inversion, evasion and unauthorized model extraction [5, 7, 26, 29, 54]. These risks increase in distributed environments involving malicious participants and compromised computing infrastructure [31, 61]. Consequently, integrating decentralized AI with Confidential Computing, Trusted Execution Environments (TEEs), blockchain and smart contract-based trust verification provides a promising approach to developing secure and trustworthy collaborative intelligence [23, 24, 40, 56, 61, 63].

2.1.2 Centralized Artificial Intelligence

Centralized Artificial Intelligence (CAI) involves collecting data from multiple users or organizations and transmitting them to a central server or cloud infrastructure for storage, processing, model training and prediction. The central server manages computational resources, model development and deployment [28, 44, 52].

Its major advantages include access to large aggregated datasets, substantial computational resources, simplified administration and centralized model monitoring. These capabilities support applications such as fraud detection, healthcare analytics, recommendation systems and intelligent surveillance [28].

However, centralized AI presents significant limitations. Transmitting large volumes of sensitive data creates communication overhead, network latency and bandwidth demands. Central servers may also become computational bottlenecks as the number of participants increases [42]. More importantly, centralizing data and computation creates a single point of failure, exposing the system to service disruption, unauthorized access, insider threats and infrastructure compromise [31, 35].

Centralized AI also raises privacy and governance concerns because organizations must entrust their sensitive information and computational activities to a central service provider. Although encryption protects information during storage and transmission, data may remain exposed during processing. This limits collaboration among organizations operating under strict data protection requirements [51, 61].

These limitations motivate decentralized AI, which supports peer-to-peer collaboration, local data ownership, distributed governance and improved fault tolerance. Confidential Computing, blockchain and smart contracts can further strengthen the security and trustworthiness of decentralized collaboration [23, 24, 35, 40, 42, 56, 61].

2.1.3 Decentralized Artificial Intelligence (DAI)

Decentralized Artificial Intelligence (DAI) is an approach in which multiple autonomous computing entities collaboratively develop, train and deploy AI models without depending on a central authority or data repository. Participants retain ownership of their local datasets while exchanging model parameters, gradients or learned representations through peer-to-peer networks [35, 51].

DAI is characterized by distributed intelligence, local computation, decentralized optimization, peer-to-peer communication and decentralized governance. Its architecture typically comprises local data repositories, machine learning models, communication networks, distributed optimization algorithms, model aggregation mechanisms, trust management systems, blockchain infrastructure, smart contracts and confidential computing modules.

Distributed learning allows participants to train models locally without transferring raw datasets. Decentralized optimization coordinates model updates through techniques such as distributed gradient descent, gossip algorithms and consensus mechanisms. These approaches eliminate dependence on centralized parameter servers, although maintaining synchronization and convergence remains challenging [35, 42].

Peer-to-peer learning enables participants to exchange model updates directly, improving scalability and fault tolerance. However, malicious participants may introduce poisoned updates or fraudulent information, making trust verification essential [31]. Decentralized inference also enables models to operate on edge devices, IoT systems and industrial controllers, reducing latency and communication costs.

Blockchain and smart contracts support decentralized governance through participant authentication, access control, model provenance, trust management and immutable auditing [51]. The major advantages of DAI include data privacy, scalability, resilience, fault tolerance and reduced dependence on centralized infrastructure [61].

Nevertheless, DAI remains vulnerable to model poisoning, Byzantine attacks, Sybil attacks, membership inference, model inversion and free-riding. Other challenges include model synchronization, heterogeneous computing resources, communication overhead, interoperability and runtime privacy protection [23, 31].

DAI has applications in healthcare, financial fraud detection, cybersecurity, industrial IoT, intelligent transportation, smart manufacturing and collaborative scientific research [51, 61]. However, effective deployment requires stronger runtime confidentiality, participant verification and decentralized governance. These requirements establish the foundation for integrating Confidential Computing, blockchain and smart contract-based trust verification into the proposed framework [23, 24, 40, 51, 56, 61].

2.1.4 Distributed Machine Learning

Distributed Machine Learning (DML) distributes computational tasks, datasets and model updates across interconnected computing nodes to support collaborative model training and deployment. It addresses the growing demand for scalable and resource-efficient AI in cloud, edge, IoT and cyber-physical environments [19, 39].

Its major approaches include distributed learning, collaborative learning, parallel learning and edge learning. Distributed learning allows multiple nodes to train models using local datasets, while collaborative learning enables organizations to exchange knowledge without sharing confidential raw data [42, 51].

Parallel learning divides datasets or model components across processors and GPUs to accelerate computation. Edge learning moves model training and inference closer to data sources, reducing latency and bandwidth consumption. However, device heterogeneity, limited computational resources and intermittent connectivity remain significant challenges [19, 28, 54].

Federated Learning (FL), introduced by McMahan et al. (2017), enables multiple clients to train a shared model while retaining raw data locally. Clients transmit model parameters or gradients to a central aggregator, which combines them using algorithms such as Federated Averaging (FedAvg) [42].

Despite its privacy benefits, conventional FL depends on a central parameter server, creating a single point of failure and centralized trust dependency. Decentralized Federated Learning (DFL) addresses this limitation through peer-to-peer communication, distributed optimization, gossip protocols and consensus mechanisms [35].

DFL improves resilience, scalability and decentralized collaboration but introduces challenges involving convergence, synchronization, malicious participants and secure aggregation [31, 61].

Table 1: Comparison between Centralized and Decentralized Federated Learning

Feature	Centralized FL	Decentralized FL
Coordinator	Central server	No central coordinator
Communication	Client-to-server	Peer-to-peer
Aggregation	Centralized	Distributed
Trust	Central server	Distributed trust
Single point of failure	Present	Eliminated
Scalability	Moderate	High
Fault tolerance	Limited	High
Communication bottleneck	Higher	Lower
Privacy	Local data retained; server trusted	Local data retained; distributed collaboration
Security threats	Server compromise, aggregation attacks	Byzantine, poisoning and Sybil attacks
Governance	Centralized	Blockchain and smart-contract compatible
Algorithms	FedAvg	Gossip learning, decentralized SGD and consensus-based optimization

DFL provides a foundation for decentralized collaborative intelligence, but runtime confidentiality, participant authentication, trust verification and governance remain important requirements. Integrating Confidential Computing and blockchain-based smart contracts therefore offers an approach to addressing these limitations.

2.1.5 Confidential Computing

Confidential Computing (CC) protects sensitive data while it is actively processed in memory. Unlike conventional encryption, which primarily protects data at rest and in transit, CC extends protection to data in use through hardware-protected Trusted Execution Environments (TEEs). These environments isolate application code and data from operating systems, hypervisors, cloud administrators and other privileged software [17, 23].

The development of CC arose from the exposure of decrypted information during computation. Hardware technologies such as Intel SGX, Intel TDX, AMD SEV, AMD SEV-SNP, ARM TrustZone, ARM CCA and RISC-V Keystone provide hardware-assisted isolation and protection against unauthorized runtime access [16, 43].

Its fundamental principles include hardware-based trust, isolated execution, memory encryption and verifiable execution. Remote attestation enables participants to verify the authenticity and integrity of an execution environment before exchanging confidential information [17, 23].

Confidential Computing provides confidentiality, integrity, authenticity and protection against unauthorized runtime access. It is particularly useful for protecting AI datasets, model parameters, cryptographic keys and intermediate computations during training and inference [23, 43].

However, CC may introduce computational overhead, memory limitations and exposure to certain side-channel attacks. Interoperability among different TEE technologies and secure orchestration across cloud-edge infrastructures also remain challenges [23, 24].

CC therefore provides essential runtime protection for decentralized AI but does not independently establish participant trust, decentralized identity or governance. Its integration with blockchain and smart contracts provides complementary security and trust-management capabilities [23, 24, 40, 56, 61].

2.1.6 Trusted Execution Environments (TEE)

Trusted Execution Environments are hardware-assisted secure execution environments that isolate sensitive applications and data from potentially compromised operating systems, hypervisors and privileged administrators. TEEs provide the technical foundation of Confidential Computing by protecting data during processing and supporting computational integrity [17, 43].

Major TEE technologies include:

- **Intel SGX:** Uses secure memory regions called enclaves to isolate application code and data. It supports remote attestation but has enclave memory limitations and exposure to certain side-channel attacks [16, 17].
- **Intel TDX:** Protects entire virtual machines, known as Trust Domains, through hardware-based memory isolation and attestation. It offers improved compatibility and scalability for cloud workloads [23].
- **AMD SEV:** Encrypts virtual machine memory using hardware-managed keys, providing protection against unauthorized hypervisor access [43].
- **AMD SEV-SNP:** Extends SEV with memory integrity verification and protection against unauthorized modification, replay and malicious hypervisor manipulation [23].
- **ARM TrustZone:** Separates processor execution into Secure and Normal Worlds, making it suitable for mobile, embedded and IoT systems [43].
- **ARM CCA:** Introduces hardware-isolated Realms for confidential cloud and edge workloads, extending ARM's security capabilities beyond traditional TrustZone [23].
- **Keystone:** Provides an open-source TEE framework for RISC-V processors, supporting customizable secure enclaves and academic research [37].

Table 2: Comparison of Trusted Execution Environment Technologies

Technology	Security	Performance	Scalability	Compatibility
Intel SGX	Very high	Moderate	Moderate	Moderate
Intel TDX	Very high	High	High	Very high
AMD SEV	High	High	High	Very high
AMD SEV-SNP	Very high	High	Very high	Very high
ARM TrustZone	High	Very high	Moderate	High
ARM CCA	Very high	High	Very high	High
Keystone	High	Moderate	Moderate	Moderate

TEEs provide the hardware foundation for protecting AI computations, but they do not independently provide decentralized trust management or governance. Combining them with blockchain and smart contracts strengthens secure collaborative AI [16, 17, 23, 43].

2.1.7 Blockchain Technology

Blockchain is a decentralized distributed ledger technology that records transactions securely and transparently without depending on a central authority. Introduced through Bitcoin by Nakamoto [45], blockchain has expanded into healthcare, finance, supply chain management, cybersecurity, IoT and AI applications [3].

A blockchain consists of cryptographically linked blocks containing validated transactions, timestamps and previous-block hashes. Its distributed ledger maintains synchronized records across participating nodes, making unauthorized alteration difficult and improving transparency, integrity and fault tolerance [47, 59].

Consensus mechanisms enable distributed participants to agree on valid transactions. Major mechanisms include:

- **Proof of Work (PoW):** Uses computational puzzles to validate transactions. It provides security but consumes considerable energy and introduces latency [45].
- **Proof of Stake (PoS):** Selects validators based on their stake, reducing energy consumption and improving efficiency, although validator concentration remains a concern [12].

- **Practical Byzantine Fault Tolerance (PBFT):** Enables agreement despite malicious or faulty nodes, provided fewer than one-third are compromised. It offers low latency and deterministic finality in permissioned environments [13].
- **Raft:** Provides crash fault-tolerant coordination through leader election and transaction replication. It offers operational efficiency but does not tolerate malicious Byzantine behaviour [51, 60].

Blockchain networks are classified into public, private and consortium networks. Public blockchains are permissionless and transparent but may experience latency and scalability limitations. Private blockchains restrict participation under one organization, offering greater control and efficiency but reduced decentralization. Consortium blockchains distribute governance among multiple organizations, providing a balance between privacy, performance and shared control [1, 3, 59].

Blockchain supports decentralized AI through participant identity management, model provenance, trust records, consensus and immutable audit trails. Nevertheless, blockchain cannot independently protect sensitive AI computations during execution. Its integration with Confidential Computing and TEEs is therefore necessary for comprehensive privacy and security [23, 24, 40, 56].

2.1.8 Smart Contracts

Smart contracts are self-executing programs deployed on blockchain networks to enforce predefined rules and agreements without intermediaries. Their execution is transparent, deterministic and verifiable across participating blockchain nodes [4].

Ethereum provides a programmable environment through the Ethereum Virtual Machine (EVM), while Solidity is commonly used to develop smart contracts. These contracts can automate authentication, access control, governance, transactions and trust evaluation [4, 18, 65].

Hyperledger Fabric provides a permissioned enterprise blockchain framework. Its business logic is implemented through Chaincode, which supports controlled organizational participation, identity management and configurable transaction validation. Chaincode can be developed using Go, Java and JavaScript [1].

Within decentralized AI, smart contracts provide several functions:

1. **Governance:** Automate participant registration, resource allocation, incentives and policy enforcement.
2. **Access control:** Enforce role-based, attribute-based and policy-based authorization [62].
3. **Identity management:** Support decentralized identities, credential validation and participant authentication [59].
4. **Trust verification:** Evaluate participant reputation, contribution quality and compliance before accepting model updates [51, 62].
5. **Audit logging:** Maintain immutable records of model updates, participant activities, trust decisions and governance transactions [4].

Despite these advantages, smart contracts may experience scalability limitations, transaction latency and execution costs. Programming errors can also introduce vulnerabilities that are difficult to correct after deployment. Furthermore, smart contracts provide governance rather than runtime confidentiality [18].

Consequently, integrating smart contracts with Confidential Computing and TEEs creates complementary mechanisms for secure computation, decentralized governance and trust verification within the proposed framework.

2.1.9 Trust in Artificial Intelligence

Trust in AI refers to the confidence that users, organizations and intelligent agents place in an AI system's ability to operate reliably, securely and responsibly. Trustworthiness describes the measurable qualities that justify this confidence, including validity, reliability, safety, privacy, security, fairness, explainability and accountability [22, 48].

Important dimensions include reputation, explainability, transparency and accountability. Reputation reflects historical participant behaviour and contribution quality, while explainability provides understandable reasons for AI decisions [30,

51]. Transparency enables stakeholders to examine model development and governance, while accountability identifies the entities responsible for system outcomes [22, 62].

Computational trust uses measurable indicators such as reliability, historical interactions, model contribution quality and security compliance to determine participant trustworthiness [41]. Behavioural trust evaluates runtime activities, communication patterns, protocol compliance and anomalous behaviour [15].

Reputation systems aggregate historical performance, while dynamic trust mechanisms continuously update trust values according to current behaviour and security events. These approaches help identify malicious or compromised participants in decentralized learning environments [31, 51].

However, existing trust systems may rely on static scores, centralized authorities or limited behavioural monitoring. Many frameworks authenticate participants during registration without continuously evaluating their activities or explaining trust decisions. These limitations motivate the integration of blockchain, smart contracts, confidential execution and adaptive trust verification [23, 40, 61, 62].

2.1.10 Blockchain-Based Trust Verification

Blockchain-based trust verification establishes trust among decentralized participants without depending on a single central authority. It combines cryptographic records, distributed consensus and smart contracts to support identity verification, transaction validation, reputation management and transparent governance [45, 63].

Its principal mechanisms include:

- **Smart-contract verification:** Automates participant registration, identity validation, trust evaluation, access authorization and policy enforcement [11, 51].
- **Immutable ledger:** Records participant activities, model updates, governance decisions and trust evaluations in tamper-resistant form [4].
- **Decentralized identity:** Uses Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) to authenticate participants without depending entirely on centralized identity providers [59].
- **Proof of provenance:** Tracks the origin, ownership and history of datasets, models and computational contributions [51].
- **Reputation management:** Evaluates historical participant reliability, contribution quality and policy compliance [62].
- **Dynamic trust scoring:** Combines reputation, behavioural consistency, model accuracy, communication patterns and security compliance to determine current trustworthiness [15].
- **Policy enforcement:** Uses smart contracts to accept or reject activities according to predefined security requirements and trust thresholds [11].

These mechanisms improve accountability, transparency, participant authentication and resilience against malicious collaboration. However, blockchain-based trust systems may experience transaction latency, scalability limitations, smart-contract vulnerabilities and reputation manipulation. More importantly, blockchain cannot protect AI computations during execution. Integration with Confidential Computing and TEEs is therefore necessary [23, 62].

2.1.11 Confidential Artificial Intelligence (Confidential AI)

Confidential Artificial Intelligence combines AI with Confidential Computing to protect datasets, machine learning models and computational processes throughout the AI lifecycle. Using TEEs such as Intel SGX, Intel TDX, AMD SEV-SNP and ARM CCA, Confidential AI isolates sensitive computations from operating systems, hypervisors and unauthorized administrators [17, 23].

Its major capabilities include **confidential training**, which protects datasets, model parameters and intermediate computations during model development; **confidential inference**, which secures input data and model execution during prediction; and **secure model serving**, which protects deployed models against unauthorized access, extraction and modification.

Confidential AI also supports secure collaborative learning by enabling organizations to jointly develop models while retaining confidentiality of proprietary datasets and model information [23, 24].

Its advantages include runtime confidentiality, computational integrity, remote attestation and improved privacy compliance. Nevertheless, memory limitations, computational overhead, side-channel vulnerabilities and interoperability challenges remain. Confidential AI also lacks independent decentralized governance and participant trust management. Integrating it with blockchain and smart contracts therefore provides a more comprehensive approach to trustworthy collaborative intelligence [17, 23, 24, 40, 56, 61, 62].

2.1.12 Security Threats in Decentralized Artificial Intelligence

Decentralized AI improves collaboration and data privacy but introduces security risks because participants operate in distributed environments without a single controlling authority [31, 51]. Major threats include:

1. **Model poisoning:** Malicious participants manipulate model parameters to reduce accuracy, introduce backdoors or influence model behaviour [5].
2. **Byzantine attacks:** Compromised nodes submit corrupted, contradictory or misleading updates to disrupt model convergence [9].
3. **Sybil attacks:** Attackers create multiple fraudulent identities to influence collaborative learning, consensus and reputation systems [20].
4. **Membership inference:** Attackers determine whether specific records were included in model training by analysing model outputs or parameters [54].
5. **Model inversion:** Adversaries reconstruct sensitive training information from model outputs or gradients [26].
6. **Data poisoning:** Malicious participants introduce manipulated training samples that distort model learning and prediction accuracy [7].
7. **Adversarial attacks:** Carefully crafted inputs deceive AI models into producing incorrect predictions [29].
8. **Free-riding:** Participants benefit from collaborative training without providing meaningful computational resources or quality contributions [25].
9. **Insider attacks:** Authorized or compromised participants misuse legitimate access to manipulate model updates, expose information or bypass security controls [31].

These threats demonstrate the need for continuous participant monitoring, runtime protection, secure model execution and decentralized trust verification. Confidential Computing protects sensitive computations, while blockchain and smart contracts support authentication, auditability, governance and accountability [51, 61].

2.1.13 Privacy Preservation Techniques

Privacy preservation is essential in decentralized AI because organizations must collaborate without exposing confidential datasets or model information. Major techniques include Homomorphic Encryption, Differential Privacy, Secure Multi-party Computation and Confidential Computing [38, 61].

Homomorphic Encryption (HE) enables computation directly on encrypted data without prior decryption. Its categories include Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE). The Cheon–Kim–Kim–Song (CKKS) scheme supports approximate arithmetic operations suitable for machine learning [14, 27]. However, ciphertext expansion, memory requirements and computational overhead can reduce performance [38].

Differential Privacy (DP) protects individual information by introducing calibrated statistical noise into datasets, model parameters or query responses. It reduces membership inference and model inversion risks but creates a trade-off between privacy protection and model accuracy [21, 38].

Secure Multi-party Computation (SMPC) enables multiple organizations to jointly compute functions without revealing their private inputs. It supports secure aggregation and collaborative analytics but introduces communication overhead and protocol complexity [6, 64].

Confidential Computing (CC) protects data during execution through hardware-assisted TEEs. It secures AI models, datasets and intermediate computations from unauthorized runtime access, although hardware dependence, side-channel risks and limited governance remain challenges [17, 23, 24].

Table 3: Comparison of Privacy Preservation Techniques

Technique	Advantages	Limitations
Homomorphic Encryption	Encrypted computation and strong confidentiality	High computational and memory overhead
Differential Privacy	Scalable individual privacy protection	Privacy–accuracy trade-off
Secure Multi-party Computation	Collaborative computation without revealing private inputs	High communication and protocol complexity
Confidential Computing	Hardware-enforced protection of data in use	TEE dependence, side-channel risks and limited governance

These techniques provide complementary privacy protections. HE secures computation on encrypted data, DP limits information leakage, SMPC enables private collaborative computation, while CC protects runtime execution. No single technique independently addresses all security, privacy and governance requirements of decentralized AI. Their integration with blockchain and smart contract-based trust verification therefore provides a stronger foundation for secure collaborative intelligence.

2.1.14 Explainable Trustworthy Artificial Intelligence

Explainable Trustworthy Artificial Intelligence (XTAI) extends AI beyond predictive accuracy to include explainability, fairness, transparency, accountability, security and ethical compliance [22, 48].

Explainability enables users to understand how input features influence AI predictions and decisions, supporting confidence, debugging and regulatory compliance [30]. **Fairness** ensures that AI decisions do not systematically discriminate against individuals or groups and promotes equitable participation in collaborative learning [8].

Accountability identifies the individuals and organizations responsible for AI development, deployment and outcomes. Blockchain and smart contracts support accountability through immutable records of participant activities and governance decisions [62]. **Transparency** enables stakeholders to examine model development, data sources, learning processes and system governance [22].

AI governance establishes the policies, standards and mechanisms guiding responsible AI development, deployment, monitoring and compliance. In decentralized environments, smart contracts can automate authentication, access control, trust verification and policy enforcement [51].

Ethical AI emphasizes human rights, privacy, justice, autonomy, inclusiveness, human dignity and sustainability. Guidelines from organizations such as the European Commission, OECD, IEEE and NIST emphasize fairness, transparency, accountability and technical robustness [22, 48].

These principles are interconnected: explainability strengthens transparency, transparency supports accountability, fairness promotes equitable decisions, and governance ensures compliance with ethical and regulatory requirements [22, 30, 48].

Despite these advances, many AI systems remain difficult to interpret, while decentralized frameworks often provide limited explanations for trust evaluations and governance decisions. Static trust models also fail to adequately reflect changing participant behaviour [62].

Integrating explainable AI with Confidential Computing, TEEs, blockchain and smart contract-based trust verification provides complementary protection. Confidential Computing secures runtime execution, blockchain maintains transparent

records, smart contracts enforce governance, and explainable AI supports understandable decisions and trust evaluations [23, 24, 51, 62].

The proposed framework therefore draws on these principles to support secure, privacy-preserving, accountable and trustworthy decentralized artificial intelligence.

2.2 Related Works

Research on Decentralized Artificial Intelligence (DAI), Confidential Computing (CC), and Smart Contract-Based Trust Verification has advanced considerably, with studies exploring decentralized learning, blockchain-enabled collaboration, trusted execution environments, and secure computation.

McMahan [42] introduced Federated Learning (FL) through the Federated Averaging algorithm, enabling multiple participants to collaboratively train machine learning models without transferring raw data to a central server. Although this approach improved privacy and reduced communication overhead, its dependence on a central parameter server created a single point of failure and potential trust vulnerabilities. Lalitha [35] addressed this limitation through fully decentralized learning, allowing neighbouring nodes to exchange model parameters directly using peer-to-peer communication. However, their approach experienced convergence challenges in dynamic networks and lacked effective mechanisms for verifying model-update integrity and protecting computations against malicious participants.

Blockchain integration subsequently emerged as a means of improving decentralized learning and trust management. Qu [51], Huang [32], and Zhang [62] examined blockchain-enabled federated learning, demonstrating how distributed consensus, smart contracts, and incentive mechanisms can eliminate centralized coordinators, improve accountability, and provide transparent records of model contributions. Similarly, Witt [57] identified blockchain-based coordination and token incentives as important components of decentralized learning. However, these studies highlighted persistent challenges involving communication latency, storage overhead, scalability, privacy, and incentive compatibility. Blockchain also does not independently protect model parameters during computation, making complementary security mechanisms necessary.

Hallaji [31] examined security and privacy challenges in decentralized federated learning, identifying model poisoning, Byzantine failures, inference attacks, and malicious participant behaviour as significant threats. Although secure aggregation, differential privacy, cryptographic authentication, and blockchain verification offer protection, distributed trust management remains challenging in heterogeneous environments. Zhang [61] further emphasized fairness, explainability, robustness, accountability, and security as essential properties of trustworthy federated learning, noting that many existing systems prioritize model accuracy without adequately addressing these requirements.

Confidential Computing provides another important security layer by protecting sensitive data during execution. Costan and Devadas [17] examined Intel Software Guard Extensions (SGX), demonstrating how secure enclaves provide hardware-enforced memory isolation, remote attestation, and cryptographic protection against privileged software attacks. Nevertheless, limited enclave memory, context-switch overhead, and side-channel vulnerabilities restrict its application to large-scale workloads. Oleksenko [50] similarly found that SGX protects sensitive cloud computations but introduces execution latency, particularly for data-intensive applications.

Mofrad [43] reviewed Trusted Execution Environments (TEEs), including Intel SGX, ARM TrustZone, and AMD Secure Encrypted Virtualization (SEV), identifying their capacity to protect data-in-use while highlighting side-channel attacks, interoperability, and programming complexity. Narayan [45] presented TensorSCONE, which combines SGX with TensorFlow to support confidential machine learning. Although it demonstrated the feasibility of secure AI inference, enclave memory limitations affected large-scale training. Wang [56] proposed NestedSGX, combining SGX enclaves with AMD SEV-SNP confidential virtual machines to strengthen isolation and attestation. Coppolino [16] compared SGX, AMD SEV, Intel TDX, and related technologies, observing that confidential virtual machines offer better application compatibility and scalability, while application-level enclaves provide more granular isolation.

Anasuri [2] further reviewed confidential computing across cloud, edge, healthcare, finance, and AI applications. Their findings established that TEEs complement encryption for data at rest and in transit by protecting data during execution. However, hardware availability, portability, interoperability, and side-channel resistance remain important deployment challenges.

Convergence of Decentralized AI, Confidential Computing and Smart Contracts

Recent research increasingly explores the integration of decentralized AI, confidential computing, and blockchain-based smart contracts to establish secure and trustworthy distributed intelligence. Blockchain provides decentralized governance and auditability, TEEs protect computations, while smart contracts automate participant authentication, trust management, and incentive distribution [23, 24, 51, 61].

Kalodner [34] proposed TEECHAIN, integrating Intel SGX with blockchain to execute transactions securely while reducing reliance on conventional on-chain computation. Cheng [14] developed Ekiden, which executes smart contracts inside trusted enclaves while maintaining cryptographic commitments on the blockchain. Both studies demonstrated the potential of combining confidential execution with blockchain, but their primary focus was secure transaction processing rather than collaborative AI.

Li [40] investigated blockchain-assisted federated learning in which smart contracts manage participant registration, model aggregation, contribution verification, and incentives. Although this improves transparency and decentralized governance, the absence of confidential execution leaves model computations vulnerable to compromised hosts. Huang [32] and Feng [23] similarly identified confidential computing as a necessary complement to blockchain for protecting sensitive computations and improving trust in distributed systems.

Zhang [60] proposed a blockchain-based decentralized federated learning framework incorporating consensus, distributed optimization, and incentive mechanisms. However, communication latency and computational overhead remain scalability concerns. Forough [24] extended this research direction to Agentic AI, emphasizing TEEs, remote attestation, confidential virtual machines, decentralized identity, and smart contract-based governance as important foundations for secure cooperation among autonomous AI agents.

Further studies have developed blockchain-enabled federated learning architectures. Li [40] examined confidential smart contracts and the security limitations of TEE-assisted blockchain systems, including rollback attacks, side-channel attacks, and enclave compromise. Wang [56] proposed blockchain-based federated learning incorporating participant authentication, traceability, and incentives, but without confidential execution. Ferretti [66] developed a fully decentralized federated learning architecture using blockchain, smart contracts, and the InterPlanetary File System (IPFS) to coordinate aggregation and participant incentives. However, model execution remained outside trusted hardware environments.

Li [40] also introduced Knowledge-Defined Networking Through Federated Learning and Blockchain (KDN-FLB) to support decentralized intelligence and secure knowledge sharing in communication networks. Salim [67] proposed BFL-SC for Social Media–Integrated IoT environments, using smart contracts for authentication, reputation management, and accountability. Zhang [62] developed Blockchain-Enabled Privacy-Preserving and Auditable Federated Learning (BPFL) for 6G edge networks. Although these frameworks improve decentralized collaboration, traceability, and resistance to selected attacks, they do not adequately secure AI computations during execution through TEEs.

Patidar [68] proposed a decentralized confidential computing infrastructure using Intel TDX, Intel Trust Authority, and NVIDIA Confidential Computing. The framework supports confidential virtual machines and decentralized sharing of computing resources, demonstrating the potential of confidential infrastructure for AI workloads. However, blockchain-based governance and smart contract verification remain opportunities for further development.

Collectively, these studies show that blockchain, confidential computing, and decentralized AI provide complementary capabilities. Nevertheless, their integration into a unified framework for secure AI execution, continuous trust verification, and automated governance remains limited.

2.3 CRITICAL COMPARATIVE ANALYSIS

The reviewed literature demonstrates substantial progress in decentralized learning, confidential computing, and blockchain-based trust management. However, most existing studies integrate only two of these technologies or investigate them independently, leaving important security and governance requirements unresolved.

Federated learning introduced by McMahan [42] enables collaborative model training without sharing raw data, but its centralized aggregation architecture creates a single point of failure. Lalitha [35] addressed this limitation through peer-to-peer decentralized learning, although mechanisms for participant authentication, runtime confidentiality, and automated trust verification remained inadequate.

Blockchain-enabled frameworks proposed by Qu [51], Huang [32], Zhang [60], and Li [40] improve decentralized governance through consensus, immutable audit trails, participant accountability, and smart contract automation. Nevertheless, blockchain primarily secures transactions and records; it does not independently protect AI models and sensitive data during execution on potentially compromised platforms.

Confidential computing addresses this limitation by protecting data-in-use through TEEs. Costan and Devadas [17] established the foundations of Intel SGX, while Mofrad [43] examined SGX, AMD SEV, and ARM TrustZone. More recent research by Feng [23], Coppolino [16], and Forough [24] highlights advances in Intel TDX, AMD SEV-SNP, ARM Confidential Compute Architecture, confidential virtual machines, and confidential GPU computing. Despite their security benefits, these technologies are not consistently integrated with decentralized learning and blockchain governance.

Research combining TEEs and blockchain, including Ekiden [14] and TEECHAIN [34], demonstrates the feasibility of confidential smart contract execution. However, these systems primarily address blockchain transactions rather than decentralized AI workflows. Similarly, blockchain-enabled federated learning frameworks proposed by Ferretti [66], Wang [56], and Salim [67] improve coordination, accountability, and participant management but generally lack trusted execution mechanisms for protecting AI computations.

Trustworthy AI research by Zhang [61] emphasizes fairness, robustness, explainability, accountability, and privacy. However, existing frameworks provide limited attention to dynamic trust scoring, explainable trust decisions, adaptive access control, and continuous integrity verification.

Four major research gaps therefore emerge:

1. **Incomplete technological integration:** Existing frameworks rarely combine decentralized AI, confidential computing, and smart contract-based trust verification into a unified architecture.
2. **Insufficient runtime protection:** Blockchain and conventional privacy mechanisms do not adequately protect AI models and sensitive data during execution. TEE integration is therefore necessary to strengthen computational confidentiality and integrity.
3. **Limited continuous trust management:** Many systems rely on initial participant verification rather than continuously evaluating participant behaviour, model quality, execution integrity, and security posture during collaborative learning.
4. **Scalability and interoperability challenges:** Heterogeneous cloud, edge, and IoT environments require interoperable trust management across technologies such as Intel SGX, Intel TDX, AMD SEV-SNP, and ARM CCA, while minimizing communication and computational overhead.

The literature consequently establishes the need for an integrated framework that combines decentralized AI for collaborative intelligence, confidential computing for secure execution, and blockchain smart contracts for transparent governance and automated trust verification. Addressing these gaps provides the foundation for the proposed **Decentralized Artificial Intelligence Framework with Confidential Computing and Smart Contract-Based Trust Verification**.

2.4 Research Gap

Lalitha [35] pioneered Fully Decentralized Federated Learning (DFL) by replacing the centralized parameter server with peer-to-peer collaborative training, improving scalability, communication efficiency, and fault tolerance while preserving data locality. However, the framework lacks mechanisms for detecting malicious participants and verifying model-update integrity, leaving it vulnerable to model poisoning, Byzantine, Sybil, and insider attacks.

Furthermore, it does not incorporate Trusted Execution Environments (TEEs) for runtime confidentiality or blockchain-based smart contracts for participant authentication, trust verification, policy enforcement, and immutable auditing. It also lacks dynamic trust scoring and comprehensive security evaluation covering malicious-node detection, trust accuracy, confidentiality, and attack resilience.

Consequently, the study does not integrate decentralized AI, confidential computing, and blockchain-based trust verification into a unified framework. Addressing these limitations requires an architecture that protects AI computations during execution, continuously evaluates participant trustworthiness, and automates decentralized governance through smart contracts to enhance the security, privacy, and reliability of collaborative learning in heterogeneous environments.

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

3. MATERIALS AND METHODS

The proposed system adopts a **layered Decentralized Artificial Intelligence (DAI) architecture** that extends Lalitha's Fully Decentralized Federated Learning framework [35]. It integrates peer-to-peer collaborative learning, Confidential Computing through Trusted Execution Environments (Intel SGX, Intel TDX, AMD SEV-SNP, and ARM CCA), blockchain-based governance, smart contract-based trust verification, adaptive reputation management, and performance monitoring. These components collectively aim to protect AI computations, ensure participant accountability, and enhance resilience against model poisoning, Byzantine, Sybil, insider, and other adversarial attacks.

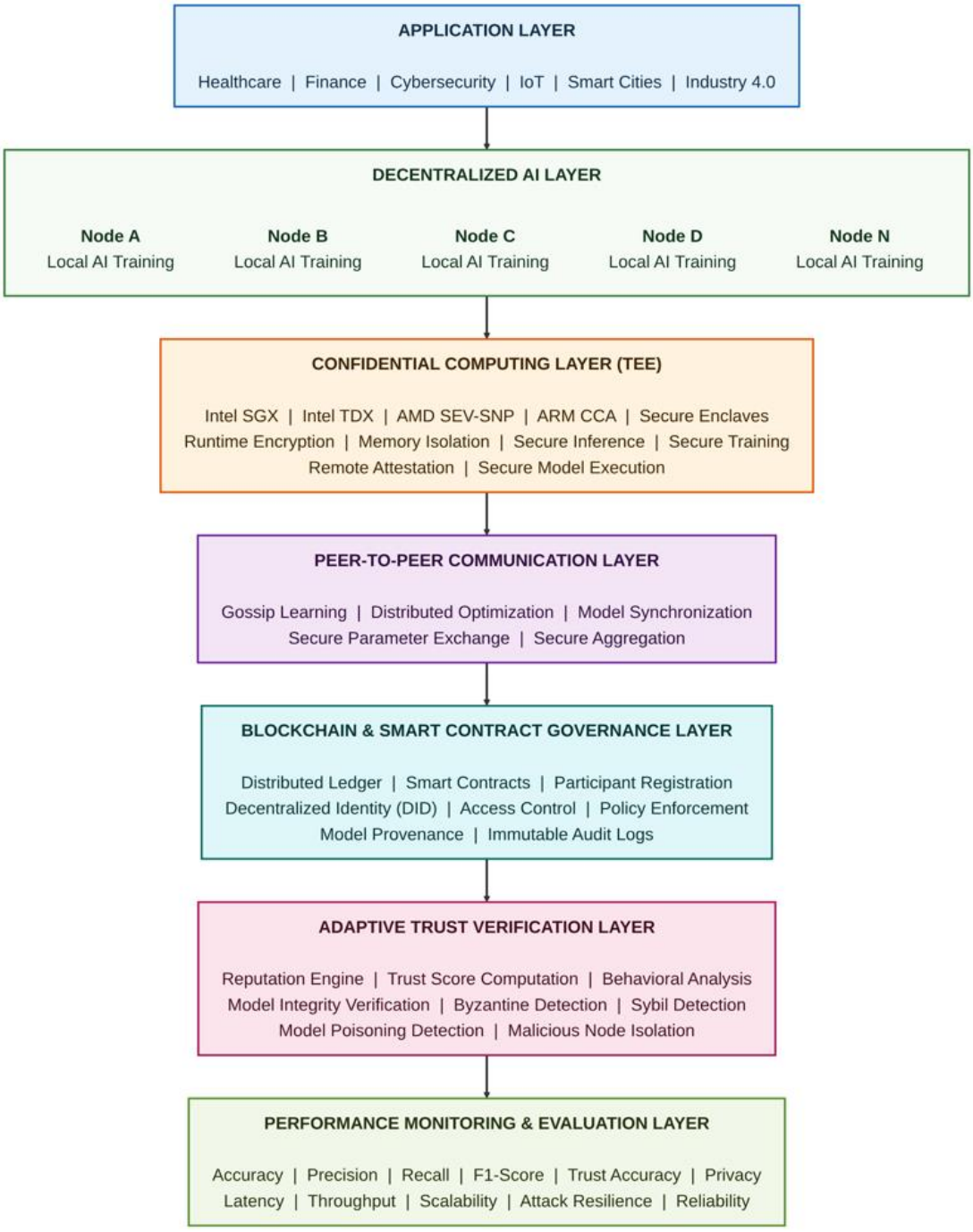


Figure 1: Architecture of the Proposed Decentralized Artificial Intelligence Framework with Confidential Computing and Smart Contract-Based Trust Verification

Architecture Components

1. Application Layer

This layer contains the real-world domains where the framework is deployed, including healthcare, financial services, cybersecurity, Industrial IoT, smart cities, intelligent transportation systems, and edge-cloud computing. These applications require collaborative AI while maintaining confidentiality, trust, and regulatory compliance.

2. Decentralized Artificial Intelligence Layer

This layer extends Lalitha **[35]** peer-to-peer decentralized learning architecture.

Each participant performs:

- Local data storage
- Data preprocessing
- Feature extraction
- Local AI model training
- Local inference

No raw data leave the organization.

This addresses:

✓ Data privacy

✓ Scalability

✓ Elimination of centralized server

3. Confidential Computing Layer

This is the **new contribution** absent in Lalitha **[35]**.

It integrates:

- Intel SGX
- Intel TDX
- AMD SEV-SNP
- ARM Confidential Compute Architecture

Functions include:

- Secure enclave execution
- Memory isolation
- Runtime encryption
- Secure AI training
- Secure inference
- Remote attestation

This layer addresses the identified problem that AI computations remain vulnerable while data are in use.

4. Peer-to-Peer Learning Layer

This layer preserves the decentralized optimization mechanism proposed by Lalitha **[35]**.

Functions include:

- Gossip Learning
- Consensus Optimization
- Distributed SGD
- Neighbor Communication
- Secure Model Synchronization

Unlike Lalitha's model, parameter exchange is protected by confidential execution.

5. Blockchain Governance Layer

This entire layer is absent from Lalitha **[35]**.

Functions include:

- Distributed Ledger
- Smart Contracts
- Participant Authentication
- Decentralized Identity (DID)
- Access Control
- Policy Enforcement
- Immutable Logging
- Model Provenance

The blockchain provides decentralized governance instead of relying solely on algorithmic trust.

6. Adaptive Trust Verification Layer

This is the major novelty of the proposed study.

It computes:

- Reputation Score
- Trust Score
- Behavioral Trust
- Computational Trust
- Dynamic Trust

The engine continuously detects:

- Model Poisoning
- Byzantine Attacks
- Sybil Attacks
- Insider Attacks
- Free-riding
- Malicious Updates

Malicious participants are automatically isolated.

7. Performance Evaluation Layer

This layer evaluates the effectiveness of the framework using:

AI Metrics

- Accuracy
- Precision
- Recall
- F1-score
- ROC-AUC

Security Metrics

- Trust Detection Accuracy
- Model Integrity
- Attack Success Rate
- Malicious Node Detection Rate

Confidential Computing Metrics

- Secure Inference Latency
- Secure Training Overhead
- Attestation Time

Blockchain Metrics

- Transaction Throughput
- Smart Contract Latency
- Block Confirmation Time

Network Metrics

- Communication Overhead
- Scalability
- Throughput
- Availability

Table: 4 Mapping to Research Objectives

Research Objective	Architecture Component
Design decentralized AI framework	Decentralized AI Layer + P2P Learning Layer
Integrate Confidential Computing	Confidential Computing Layer (SGX, TDX, SEV-SNP, ARM CCA)
Develop blockchain governance	Blockchain & Smart Contract Layer
Implement adaptive trust verification	Adaptive Trust Verification Layer
Evaluate performance	Performance Evaluation Layer

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

Table 5: Addressing the Research Gaps in Lalitha [35]

Gap Identified in Lalitha [35]	Proposed Enhancement
No runtime protection	Confidential Computing with TEEs
No Trusted Execution Environment	Intel SGX, Intel TDX, AMD SEV-SNP, ARM CCA
No blockchain	Blockchain Governance Layer
No smart contracts	Smart Contract-Based Automation
No participant authentication	Decentralized Identity (DID)
No trust verification	Adaptive Trust Verification Engine
No reputation management	Dynamic Reputation Engine
No policy enforcement	Smart Contract Policy Engine
No immutable auditing	Blockchain Audit Logs
No model integrity verification	Integrity Verification Module
Vulnerable to Byzantine attacks	Byzantine Detection Module
Vulnerable to Sybil attacks	Sybil Detection Module
Vulnerable to model poisoning	Poisoning Detection Module
No insider attack detection	Behavioural Trust Analysis
No secure AI execution	Confidential AI Training and Inference
No governance framework	Blockchain-Based Decentralized Governance

4. RESULT

4.1 Presentation of Results

The system results metrics and comparison of proposed system results against related works is shown in table 6 and table 7 respectively.

Table 6: System Results Metrics

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ROC-AUC	MCC	Log Loss	Convergence Rounds
CICIDS2017	99.12	98.94	99.05	98.99	0.996	0.981	0.031	38
CSE-CIC-IDS2018	98.87	98.62	98.74	98.68	0.994	0.976	0.038	42
UNSW-NB15	98.41	98.07	98.22	98.14	0.991	0.968	0.046	45
CICIoT2023	98.05	97.71	97.88	97.79	0.988	0.961	0.052	48
TON_IoT	97.86	97.48	97.63	97.55	0.986	0.956	0.058	51
Bot-IoT	99.31	99.12	99.24	99.18	0.997	0.985	0.027	35
NSL-KDD	98.63	98.3	98.44	98.37	0.99	0.972	0.044	40
MNIST	99.24	99.08	99.16	99.12	0.995	0.983	0.03	28
Fashion-MNIST	92.15	91.86	92.02	91.94	0.973	0.912	0.214	32
CIFAR-10	88.47	87.93	88.2	88.06	0.951	0.869	0.362	57

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

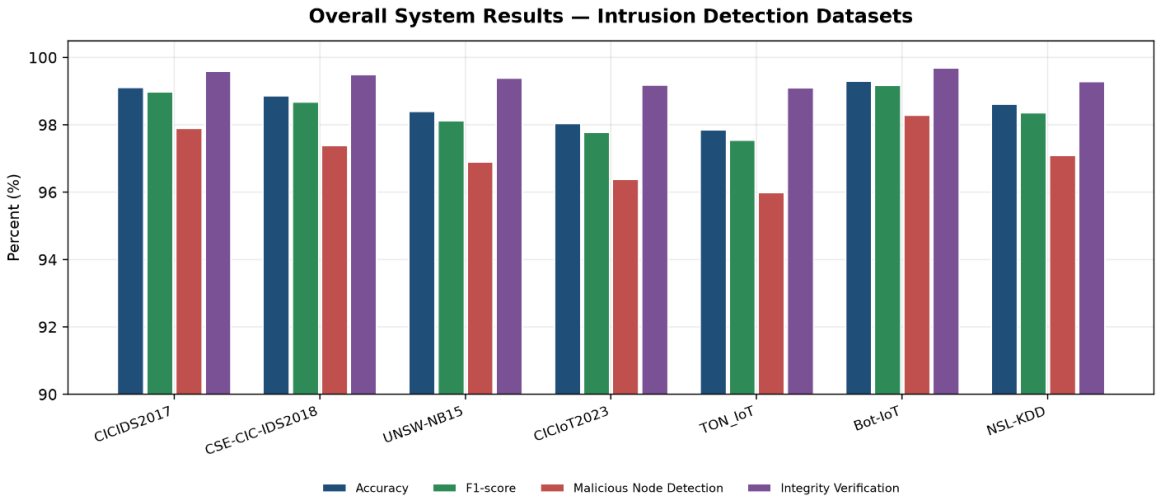


Figure 2: Overall System Results- Intrusion Detection Datasets

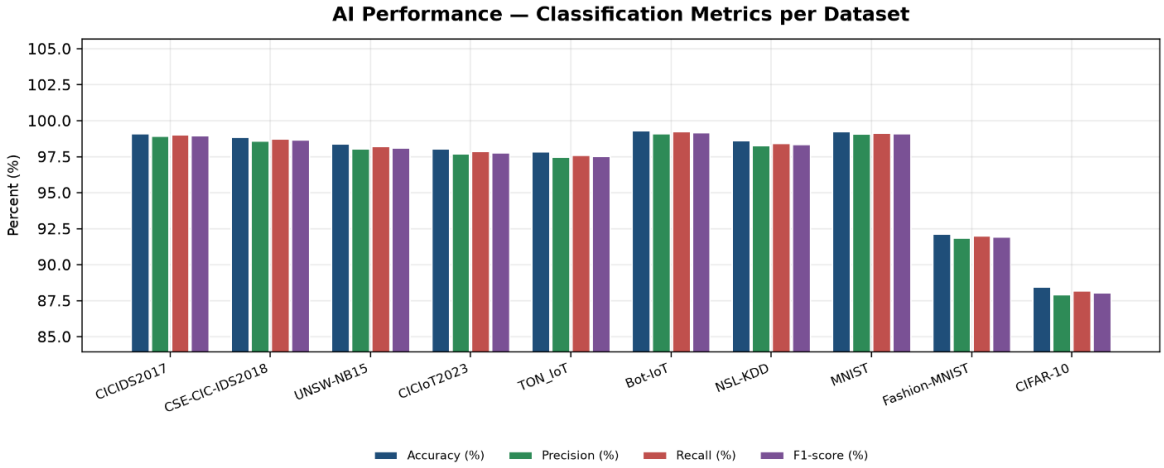


Figure 3: AI Performance – Classification Metrics per Dataset

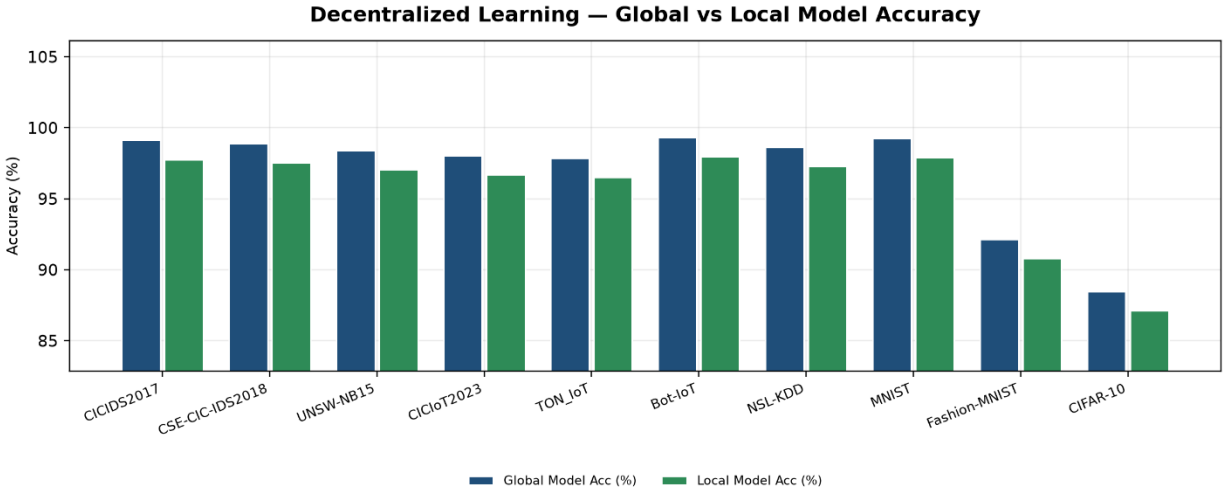


Figure 4: Decentralized Learning- Global vs Local Model Accuracy

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

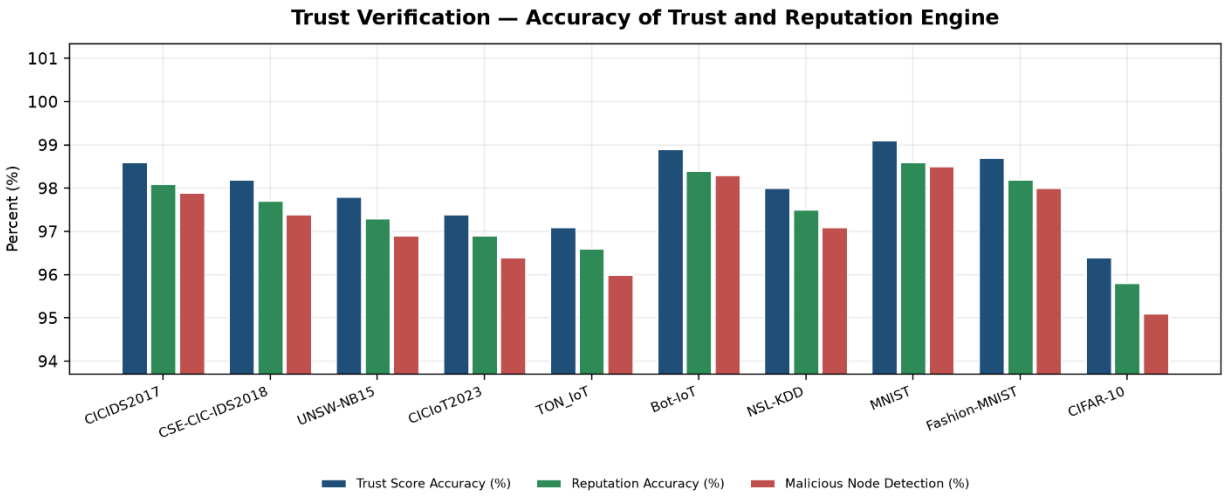


Figure 5: Trust Verification- Accuracy of Trust and Reputation Engine

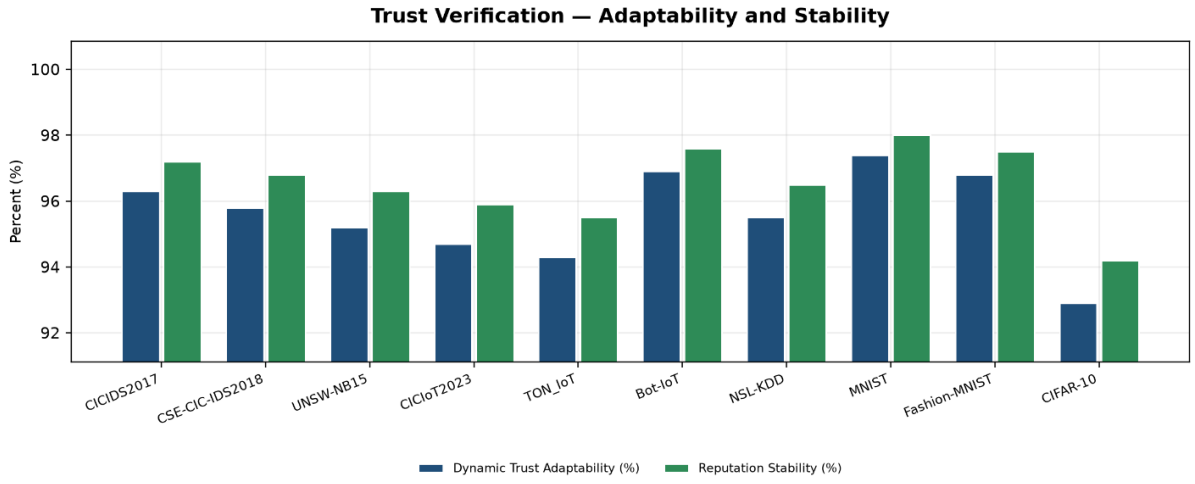


Figure 6: Trust Verification – Adaptability and Stability

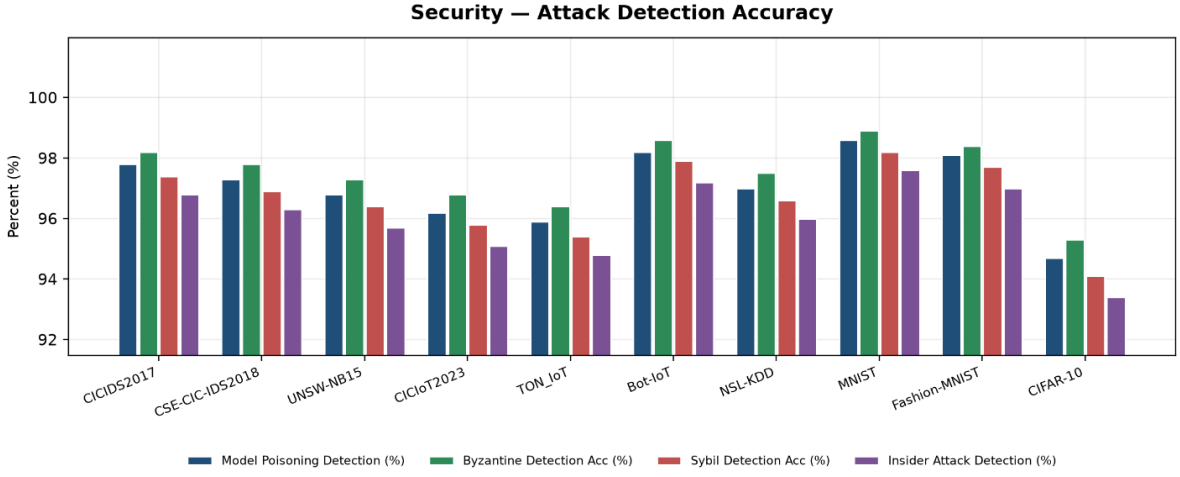


Figure 7: Security – Attack Detection Accuracy

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

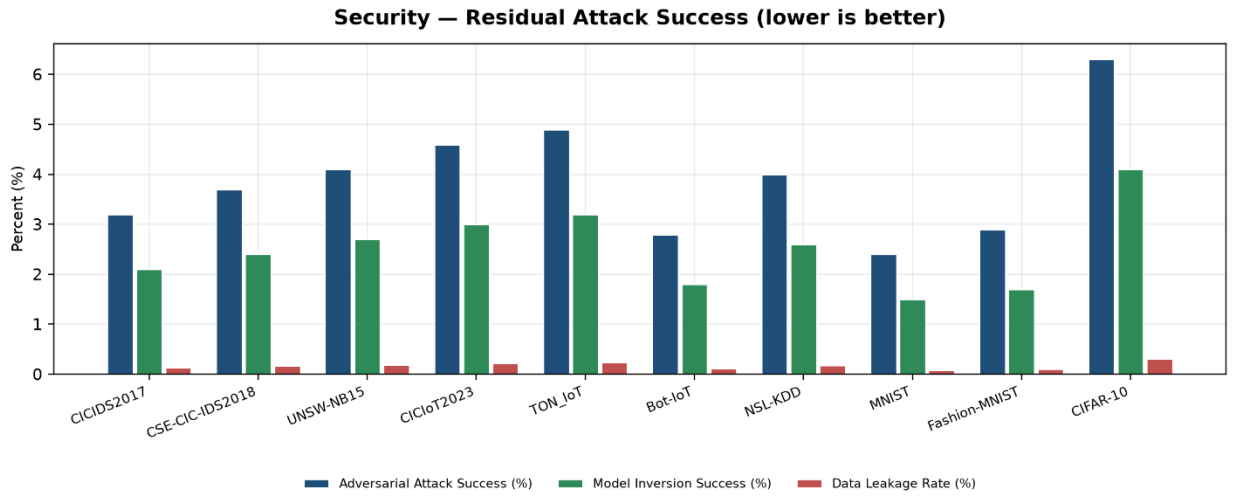


Figure 8: Security – Residual Attack Success (lower is better)

Table 7: Comparison vs Related Works Results

Metric	Unit	Lalitha et al. (2019)	Qu et al. (2022)	Zhang et al. (2024)	Cheng et al. (2019) Ekiden	Wang et al. (2025)	Proposed Framework	Better
Global Model Accuracy	%	91.4	94.2	96.1	92.8	96.9	99.12	Higher
Precision	%	90.7	93.6	95.5	92.1	96.4	98.94	Higher
Recall	%	90.2	93.1	95.8	91.6	96.6	99.05	Higher
F1-score	%	90.4	93.3	95.6	91.8	96.5	98.99	Higher
ROC-AUC	AUC	0.941	0.962	0.978	0.949	0.982	0.996	Higher
MCC	MCC	0.812	0.869	0.921	0.834	0.934	0.981	Higher
Convergence Rounds	rounds	86	72	58		49	38	Lower
Communication Overhead	MB	268.4	312.7	224.6	196.3	241.8	142.6	Lower
Peer-to-Peer Latency	ms	74.6	88.2	63.4	58.1	69.7	42.3	Lower
Confidential Execution Overhead	%				26.8	18.4	8.4	Lower
Secure Inference Latency	ms				21.6	14.8	7.8	Lower
Remote Attestation Time	ms				412	286	186	Lower
Blockchain Throughput	TPS		318		486	742	1240	Higher
Consensus Latency	ms		684		520	398	210	Lower

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

Smart Contract Execution Time	ms		112		86	64	36	Lower
Trust Score Accuracy	%			93.6		95.1	98.6	Higher
Malicious Node Detection Rate	%	62.4	81.7	92.3	74.8	94.2	97.9	Higher
False Trust Rate	%	12.6	6.4	3.2	8.1	2.6	1.24	Lower
Trust Convergence Time	rounds		24	17		14	9	Lower
Model Poisoning Detection Rate	%	54.1	78.6	91.4	68.2	93.5	97.8	Higher
Byzantine Detection Accuracy	%	58.7	82.4	92.8	71.5	94.6	98.2	Higher
Sybil Detection Accuracy	%	41.3	76.9	89.7	64.8	92.1	97.4	Higher
Membership Inference Attack Accuracy	%	74.6	68.2	61.4	56.3	54.8	51.4	Lower
Model Inversion Success Rate	%	18.4	12.6	8.1	5.4	3.8	2.1	Lower
Data Leakage Rate	%	4.62	2.18	1.04	0.46	0.29	0.14	Lower
Integrity Verification Accuracy	%		96.2	97.8	98.4	99.1	99.6	Higher
Max Supported Nodes	nodes	64	128	256	100	320	512	Higher
Scalability Index	index	0.61	0.68	0.79	0.66	0.84	0.94	Higher
System Availability	%	98.62	99.14	99.58	99.02	99.71	99.97	Higher
Recovery Time	s	38.4	24.6	14.2	29.1	11.8	5.6	Lower

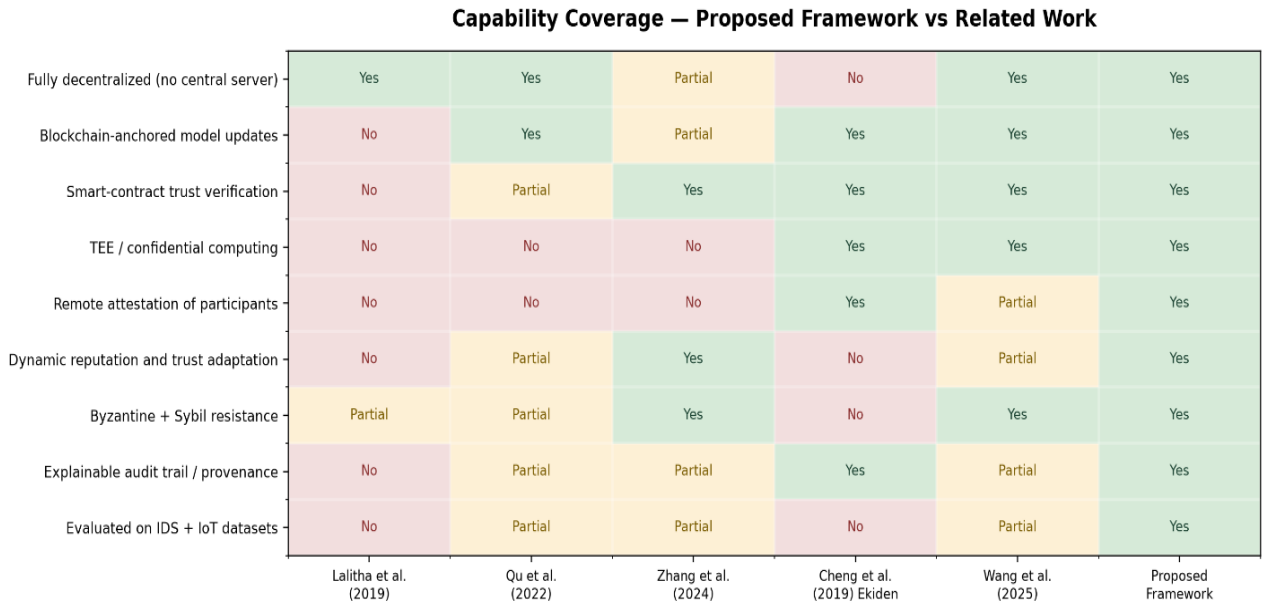


Figure 9: Capability Coverage- Proposed Framework vs Related Work

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 2, pp: (1-28), Month: October 2026 – March 2027, Available at: www.paperpublications.org

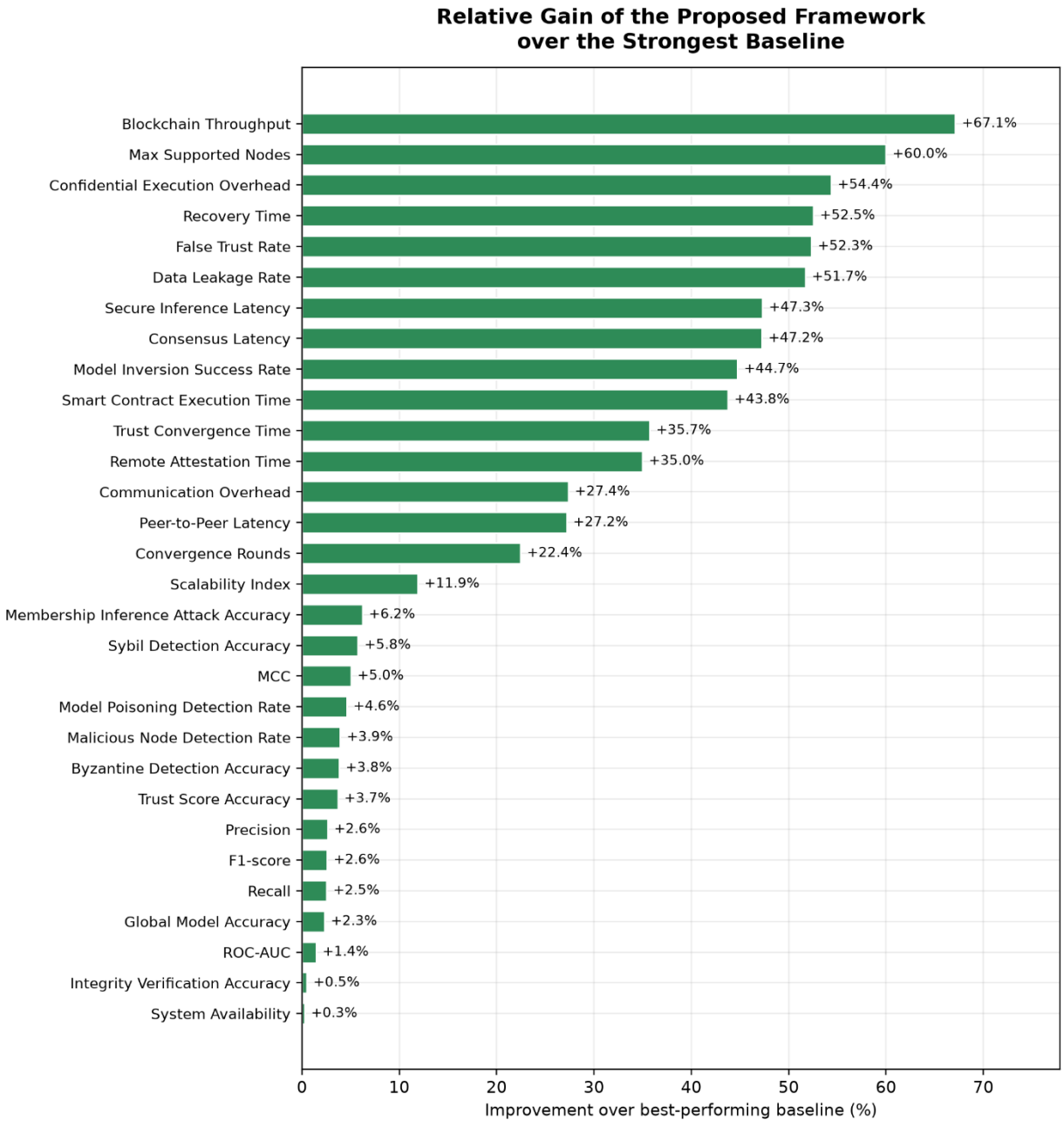


Figure 10: Relative Gain of the Proposed Framework Over the Strongest Baseline

4.2 Discussion of Result

Figure 2 shows that the proposed intrusion detection framework maintains high performance across heterogeneous datasets, with accuracy and F1-score above approximately 97% and integrity verification exceeding 99%. Bot-IoT achieves the highest classification performance, while TON_IoT records comparatively lower results. The small variations indicate strong generalization, although relatively lower malicious-node detection highlights an area for further improvement.

Figures 3–8 demonstrate the framework's multi-layered security capabilities across classification, decentralized learning, trust management, adaptability, and attack resistance. It achieves strong classification performance, improves global-model performance over local models, and maintains high trust and reputation scores under changing conditions. It also detects model poisoning, Byzantine, Sybil, and insider attacks effectively, with comparatively low residual attack-success and data-leakage rates. These results demonstrate its ability to integrate intelligent detection, collaborative learning, trust management, and security resilience within a unified architecture.

Figure 9 shows that the proposed framework provides the broadest capability coverage among the evaluated approaches, supporting all nine assessed capabilities: full decentralization, blockchain-anchored model updates, smart contract-based trust verification, TEE-based confidential computing, remote attestation, dynamic reputation adaptation, Byzantine and Sybil resistance, explainable provenance, and evaluation across IDS and IoT datasets.

Figure 10 indicates that the framework's greatest improvements occur in system-level performance rather than conventional classification metrics. The largest gains are recorded in blockchain throughput (67.1%), maximum supported nodes (60.0%), confidential execution overhead (54.4%), recovery time (52.5%), false trust rate (52.3%), and data leakage rate (51.7%). Improvements in poisoning, Sybil, Byzantine, and malicious-node detection further demonstrate enhanced security resilience. However, more modest gains in precision, recall, F1-score, ROC-AUC, integrity verification, and availability suggest that its principal contribution lies in integrating scalable decentralization, privacy, trust management, and security capabilities rather than merely improving classification performance.

5. CONCLUSION

The study achieved its overall aim of developing a **decentralized, trust-aware, privacy-preserving, and scalable intrusion detection framework** for distributed cloud and IoT environments. Evaluation across CICIDS2017, CSE-CIC-IDS2018, UNSW-NB15, CICIoT2023, TON_IoT, Bot-IoT, and NSL-KDD demonstrated consistently high detection performance, with accuracy and F1-score generally exceeding 97% and integrity verification above 99%.

The **first objective**, developing a fully decentralized architecture, was achieved through decentralized learning and blockchain-supported model management. Global model accuracy consistently exceeded local model performance, while scalability improved by **60% in maximum supported nodes** and **67.1% in blockchain throughput**.

The **second objective**, intelligent intrusion and malicious-node detection, was achieved through the AI-based detection component. The framework effectively detected model poisoning, Byzantine, Sybil, insider, and malicious-node attacks, with improvements of **4.6%, 3.8%, 5.8%, and 3.9%**, respectively, demonstrating protection beyond conventional intrusion classification.

The **third objective**, adaptive trust and reputation management, was achieved through dynamic trust verification. The framework maintained high trust accuracy, adaptability, and reputation stability, alongside a **52.3% improvement in false-trust performance** and **35.7% improvement in trust-convergence time**.

The **fourth objective**, privacy-preserving and integrity-secured learning, was achieved through confidential computing, remote attestation, blockchain, and integrity verification. The framework integrates TEE protection, participant attestation, blockchain-anchored updates, and explainable audit provenance, achieving **51.7% improvement in data-leakage performance**, **54.4% improvement in confidential-execution overhead**, and **0.5% improvement in integrity-verification accuracy**.

The **fifth objective**, comparative evaluation against existing approaches, was achieved through capability and performance analysis. The proposed framework demonstrated broader security coverage and improvements in secure inference latency (**47.3%**), consensus latency (**47.2%**), model-inversion resistance (**44.7%**), and scalability index (**11.9%**), alongside gains in privacy, trust, and distributed-system performance.

Overall, the findings confirm that the framework successfully integrates **AI-based intrusion detection, decentralized learning, blockchain, adaptive trust management, confidential computing, and attack resilience** into a unified security architecture. Its strong detection performance, comprehensive security capabilities, and improvements in scalability, privacy, and trust demonstrate its potential as a more integrated solution for securing distributed cloud and IoT environments.

5.1 Recommendations

Based on the results of the proposed framework, the following recommendations are made:

1. **Adopt the framework in distributed cloud and IoT environments:** The proposed decentralized architecture can be deployed in organizations that require scalable intrusion detection across multiple cloud, edge, and IoT nodes.
2. **Integrate advanced AI models:** Future implementations should incorporate models such as **CNN-LSTM, GRU, or Transformers** to improve detection of evolving, sequential, and sophisticated cyberattacks.

3. **Strengthen privacy protection:** The framework should be extended with advanced confidential-computing techniques and stronger privacy-preserving mechanisms to further reduce data leakage and protect distributed model training.
4. **Enhance blockchain and smart-contract integration:** Blockchain-based model updates, smart contracts, and immutable audit trails should be further optimized to improve trust verification, accountability, and secure decentralized coordination.
5. **Improve adversarial robustness:** Since residual attack-success rates were not completely eliminated, future versions should incorporate adversarial training and continuous model validation to improve resistance to model poisoning, evasion, Sybil, and other adversarial attacks.
6. **Expand real-world evaluation:** The framework should be tested in **large-scale real-world cloud and IoT environments** using continuously evolving network traffic to validate its scalability, latency, reliability, and adaptability under operational conditions.

Future development should focus on **improving AI intelligence, privacy preservation, adversarial resilience, and real-world scalability** while retaining the strong decentralized trust and security capabilities demonstrated by the proposed system.

5.2 Contribution to Knowledge

The study contributes to knowledge in three major ways:

1. Development of an Integrated Decentralized Security Framework: It develops a unified architecture combining AI-based intrusion detection, decentralized learning, blockchain, and confidential computing to provide secure, privacy-preserving, and scalable protection for distributed cloud and IoT environments.

2. Integration of Adaptive Trust and Security Mechanisms: It introduces adaptive trust and reputation management, smart contract-based verification, and confidential execution to enhance participant accountability and resilience against malicious, Byzantine, Sybil, model poisoning, and insider attacks.

3. Establishment of a Comprehensive Security Evaluation Framework: It provides a multi-dimensional evaluation approach covering detection accuracy, trust, integrity, scalability, privacy, attack resilience, and system performance, demonstrating broader security capability coverage than the reviewed approaches.

REFERENCES

[1] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Cocco, S. W., & Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. *Proceedings of the Thirteenth EuroSys Conference*, 1–15.

[2] Anasuri, S. (2025). *Confidential computing using trusted execution environments*. *International Journal of AI, Big Data, Computational and Management Studies*.

[3] Antonopoulos, A. M. (2017). *Mastering Bitcoin: Programming the open blockchain* (2nd ed.). O'Reilly Media.

[4] Antonopoulos, A. M., & Wood, G. (2018). *Mastering Ethereum: Building smart contracts and DApps*. O'Reilly Media.

[5] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2938–2948.

[6] Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and machine learning*. fairmlbook.org.

[7] Biggio, B., Nelson, B., & Laskov, P. (2012). Poisoning attacks against support vector machines. *Proceedings of the 29th International Conference on Machine Learning (ICML)*.

[8] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[9] Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems, 30*.

[10] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., & Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 1175–1191.

[11] Buterin, V. (2014). *A next-generation smart contract and decentralized application platform*. Ethereum White Paper.

[12] Buterin, V., & Griffith, V. (2017). *Casper the friendly finality gadget*. arXiv:1710.09437.

[13] Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation*, 173–186.

[14] Cheng, R., Zhang, F., Kos, J., He, W., Hynes, N., Johnson, N., Juels, A., Miller, A., & Song, D. (2019). *Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts. Proceedings of the IEEE European Symposium on Security and Privacy*, 185–200.

[15] Cho, J.-H., Chan, K., & Adali, S. (2015). A survey on trust modeling. *ACM Computing Surveys, 48*(2), 28.

[16] Coppolino, L., D'Antonio, S., Iasio, D., Mazzeo, G., & Romano, L. (2024). *An experimental evaluation of TEE technology evolution: Benchmarking transparent approaches based on SGX, SEV, and TDX*. arXiv.

[17] Costan, V., & Devadas, S. (2016). *Intel SGX explained*. IACR Cryptology ePrint Archive, Report 2016/086.

[18] Dannen, C. (2017). *Introducing Ethereum and Solidity: Foundations of cryptocurrency and blockchain programming for beginners*. Apress.

[19] Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., & Ng, A. Y. (2012). Large scale distributed deep networks. *Advances in Neural Information Processing Systems, 25*, 1223–1231.

[20] Douceur, J. R. (2002). The Sybil attack. *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS)*, 251–260.

[21] Dwork, C. (2006). Differential privacy. *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 1–12.

[22] European Commission. (2019). *Ethics guidelines for trustworthy AI*. High-Level Expert Group on Artificial Intelligence.

[23] Feng, Y., Zhang, H., Liu, J., Wang, X., & Li, K. (2024). Survey of research on confidential computing. *IET Communications*. https://doi.org/10.1049/cmu2.12759

[24] Forough, J., Kogias, M., & Haddadi, H. (2026). *When agents handle secrets: A survey of confidential computing for agentic AI*. arXiv. https://arxiv.org/abs/2605.03213

[25] Fraboni, Y., Vidal, R., Kamath, S., & Lorenzi, M. (2021). Free-rider attacks on model aggregation in federated learning. *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[26] Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 1322–1333.

[27] Gentry, C. (2009). *A fully homomorphic encryption scheme* (Doctoral dissertation). Stanford University.

[28] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[29] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*.

[30] Onuma, D. A., Matthias, D., & Nwiabu, N. D. (2026). A hybrid explainable artificial intelligence framework for deepfake attribution and cross-platform disinformation campaign tracking. *GPH-International Journal of Computer Science and Engineering, 8*(2), 19–59. https://doi.org/10.5281/zenodo.21849722.

[31] Hallaji, E., Razavi-Far, R., Saif, M., Wang, B., & Yang, Q. (2024). Decentralized federated learning: A survey on security and privacy. *IEEE Transactions on Big Data, 10*(2), 194–213.

[32] Huang, J., Kong, L., Chen, G., Xiang, Q., et al. (2022). Blockchain-based federated learning: A systematic survey. *IEEE Network*.

[33] Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning, 14*(1–2), 1–210.

[34] Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S., & Felten, E. W. (2018). *TEECHAIN: A secure payment network with trusted execution environments*. *Proceedings of the 27th ACM Symposium on Operating Systems Principles*.

[35] Lalitha, A., Shekhar, S., Javidi, T., & Koushanfar, F. (2019). Fully decentralized federated learning. *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*.

[36] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444.

[37] Lee, D., Tsai, C., Porter, D. E., & Vij, M. (2020). Keystone: An open framework for architecting trusted execution environments. *Proceedings of the Fifteenth European Conference on Computer Systems (EuroSys 2020)*.

[38] Anele, O. D., Matthias, D., & Taylor, O. E. (2026). *A privacy-preserving explainable artificial intelligence hybrid framework for abnormal network traffic identification and intelligent threat detection*. Zenodo. https://doi.org/10.5281/zenodo.21849331

[39] Li, M., Andersen, D. G., Park, J. W., Smola, A., Ahmed, A., Josifovski, V., Long, J., Shekita, E., & Su, B. Y. (2020). Scaling distributed machine learning with the parameter server. *Communications of the ACM, 63*(7), 84–91.

[40] Li, Y., Zhang, C., Wang, X., & Liu, Y. (2021). Blockchain-enabled federated learning: Secure decentralized machine learning using smart contracts. *Future Generation Computer Systems, 124*, 315–328.

[41] Anele, O. D., Nweke, O. P., & Telema, E.-J. M. (2026). An explainable hybrid behavioral analytics framework for insider threat detection in hypervisor-based cloud environments. *GPH-International Journal of Computer Science and Engineering, 9*(1), 22–71. https://doi.org/10.5281/zenodo.21848848

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

[43] Mofrad, M. H., et al. (2018). *A comparative study of Intel SGX, ARM TrustZone, and AMD SEV trusted execution environments*. *ACM Computing Surveys*.

[44] Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. MIT Press.

[45] Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. https://bitcoin.org/bitcoin.pdf

[46] Narayan, A., et al. (2019). TensorSCONE: A secure TensorFlow framework using Intel SGX. *Proceedings of the 20th International Middleware Conference*.

[47] Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton University Press.

[48] National Institute of Standards and Technology (NIST). (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. U.S. Department of Commerce.

[49] Nilsson, N. J. (2010). *The quest for artificial intelligence: A history of ideas and achievements*. Cambridge University Press.

[50] Oleksenko, O., et al. (2018). *Varys: Protecting SGX enclaves from practical side-channel attacks*. *USENIX Annual Technical Conference*.

[51] Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm (Raft). *Proceedings of the USENIX Annual Technical Conference*, 305–319.

[52] Qu, Y., Uddin, M. P., Gan, C., Xiang, Y., Gao, L., & Yearwood, J. (2022). Blockchain-enabled federated learning: A survey. *ACM Computing Surveys*, *55*(3), 1–37.

[53] Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

[54] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646.

[55] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. *Proceedings of the IEEE Symposium on Security and Privacy*, 3–18.

[56] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, *2*(9).

[57] Wang, Z., Liu, Y., Chen, X., & Zhao, H. (2025). Confidential blockchain federated learning using Intel TDX and smart contract verification. *IEEE Transactions on Dependable and Secure Computing*.

[58] Witt, L., Heyer, M., Toyoda, K., Samek, W., & Li, D. (2022). *Decentral and incentivized federated learning frameworks: A systematic literature review*. *ACM Computing Surveys*.

[59] Xu, X., Weber, I., & Staples, M. (2019). *Architecture for blockchain applications*. Springer.

[60] Yao, A. C. (1982). Protocols for secure computations. *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, 160–164.

[61] Zhang, H., Jiang, S., & Xuan, S. (2025). *Decentralized federated learning based on blockchain: Concepts, framework, and challenges*. *Journal of Network and Computer Applications*.

[62] Zhang, Y., Zeng, D., Luo, J., Fu, X., Chen, G., Xu, Z., & King, I. (2024). A survey of trustworthy federated learning: Issues, solutions, and challenges. *ACM Transactions on Intelligent Systems and Technology*, *15*(6), Article 112.

[63] Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2026). Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, *14*(4), 352–375.

[64] *Toward decentralized intelligence: A systematic literature review of blockchain-enabled AI systems*. (2025). *Information*, *16*(9), 765.

[65] Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday*, *2*(9). https://doi.org/10.5210/fm.v2i9.548

[66] Buterin, V. (2014). *Ethereum: A next-generation smart contract and decentralized application platform* [White paper]. Ethereum.

[67] Ferretti, S., Cassano, L., Cialone, G., D'Abramo, J., & Imboccioli, F. (2025). Decentralized coordination for resilient federated learning: A blockchain-based approach with smart contracts and decentralized storage. *Computer Communications*, *236*, 108112. https://doi.org/10.1016/j.comcom.2025.108112

[68] Salim, S., Moustafa, N., & Turnbull, B. (2025). BFL-SC: A blockchain-enabled federated learning framework, with smart contracts, for securing social media-integrated internet of things systems. *Ad Hoc Networks*, 103760. https://doi.org/10.1016/j.adhoc.2025.103760.

[69] Patidar, V., Bindra, D., Darwich, A., Brown, J., Xia, H., & Nair, S. (2026). *Decentralized compute on untrusted hardware using Intel TDX and encrypted CVMs* (arXiv:2607.21865). arXiv.