ICIK
Journal of Reliable and Secure Computing
http://dx.doi.org/10.62762/JRSC.2025.761390

RESEARCH ARTICLE

Blockchain Consensus Mechanisms and Enhancement Techniques for Federated Learning-Based Intrusion Detection Systems in IoT Smart Homes

Amro Alghamdi[1] **and Ismail Keshta**[1,*]

[1] Department of Applied Sciences, College of Science and Information Systems, AlMaarefa University, Riyadh, Saudi Arabia

Abstract

The rapid proliferation of smart home IoT devices has introduced unprecedented cybersecurity vulnerabilities, necessitating scalable and privacy-preserving intrusion detection systems (IDS). Federated Learning (FL) offers a promising decentralized approach by training models locally without sharing raw data, but it remains susceptible to poisoning attacks and relies on a vulnerable central aggregator. This paper presents a novel blockchain-enhanced FL framework tailored for smart home IDS, integrating multiple consensus mechanisms—Proof-of-Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), and Proof-of-Authority (PoA)—for the first time in this context. Our approach uniquely combines differential privacy (DP) and secure aggregation (SA) within a blockchain-managed workflow to mitigate gradient inversion and membership inference attacks while ensuring tamper-resistant, decentralized trust. Experimental evaluation using the N-BaIoT dataset demonstrates that the proposed system achieves up to 88.3% detection accuracy with manageable latency (200 ms/round) and formal privacy guarantees (ε=1.0 DP). The framework introduces 52.8% system overhead compared to vanilla FL—a reasonable trade-off for enhanced security and privacy. This work establishes a robust, transparent, and scalable security infrastructure for smart homes, effectively addressing the limitations of both centralized and conventional FL-based IDS.

Keywords: smart home security, internet of things (IoT), intrusion detection system (IDS), federated learning (FL), blockchain, consensus mechanisms, privacy-preserving machine learning, decentralized trust, model poisoning, cyber security.

1 Introduction

The proliferation of Internet of Things (IoT) devices within smart homes has revolutionized modern living, offering unparalleled convenience, automation, and interconnectivity. However, this rapid adoption has dramatically expanded the attack surface for cybercriminals. The global IoT ecosystem now exceeds 15 billion connected devices, with projections

Submitted: 29 December 2025
Accepted: 20 January 2026
Published: 28 January 2026

Vol. 2, **No.** 1, 2026.
10.62762/JRSC.2025.761390

***Corresponding author:**
Ismail Keshta
imohamed@um.edu.sa

Citation
Alghamdi, A., & Keshta, I. (2026). Blockchain Consensus Mechanisms and Enhancement Techniques for Federated Learning-Based Intrusion Detection Systems in IoT Smart Homes. *Journal of Reliable and Secure Computing*, 2(1), 1–26.

nearing 75 billion by 2025 [1, 2]. Each smart device—from security cameras and voice assistants to smart locks and thermostats—collects and processes highly sensitive personal data, making residential networks prime targets for exploitation. High-profile incidents, such as the Mirai botnet attacks that harnessed vulnerable IoT devices for large-scale DDoS assaults, and recurring breaches of baby monitors and home security cameras, underscore the severe and immediate risks to both personal safety and privacy [4, 5]. Furthermore, regulatory frameworks such as the GDPR and CCPA impose strict data protection obligations, increasing the urgency of privacy-preserving security solutions that operate within the constrained resources of smart home environments.

Traditional, centralized intrusion detection systems (IDS) are ill-suited to address these challenges. They create single points of failure, necessitate the continuous transmission of sensitive data to remote servers—raising critical privacy concerns—and impose unsustainable bandwidth and computational burdens on resource-constrained IoT devices. Federated Learning (FL) has emerged as a promising alternative, enabling collaborative model training across distributed devices without exchanging raw data [6]. While FL mitigates data privacy risks at the source, its standard architecture remains vulnerable [7, 8]. It relies on a centralized aggregator—a trust bottleneck and a single point of failure—and is susceptible to model poisoning, gradient inversion, and membership inference attacks [9–11]. Recent research [12] has explored blockchain as a means to decentralize trust in FL, but existing approaches often lack a holistic integration tailored for smart home constraints.

Prior works on blockchain-assisted FL for IoT security have three primary shortcomings: (1) they typically evaluate only a single consensus mechanism (e.g., PoW or PoS) without a comparative analysis of their suitability for heterogeneous smart home networks; (2) they insufficiently address the compound privacy threats in FL (e.g., combining differential privacy with secure aggregation under a blockchain-enforced workflow); and (3) they lack practical, adaptive frameworks that consider real-world deployment factors such as device heterogeneity, dynamic membership, and varying network conditions. To bridge these gaps, this paper introduces a comprehensive blockchain-enhanced FL framework designed explicitly for intrusion detection in smart home IoT. The core contributions of this work are summarized as follows:

- **A Novel Adaptive Framework:** We propose the first FL-IDS framework that dynamically integrates three distinct blockchain consensus mechanisms—Proof-of-Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), and Proof-of-Authority (PoA)—with selection strategies optimized for different smart home deployment scenarios (public, private, and consortium).

- **Enhanced Privacy-Preserving FL Protocol:** Our design uniquely combines Differential Privacy (DP) and Secure Aggregation (SA) within a blockchain-managed workflow, providing mathematically bounded privacy guarantees (ε-DP) and cryptographic protection against gradient-based inference attacks, all enforced and verified via smart contracts.

- **Comprehensive System Design:** We introduce a modular architecture featuring hierarchical RBAC via smart contracts, Computational Volume Weighting (CVW) for fair contribution, and a multi-stage validation pipeline to ensure model quality and resist poisoning attacks.

- **Extensive Empirical Validation:** We conduct rigorous experiments on the N-BaIoT dataset, demonstrating that our framework achieves competitive detection accuracy (up to 88.3%) with acceptable latency (200 ms/round) while introducing manageable system overhead (+52.8% vs. vanilla FL). A detailed analysis of the privacy-utility trade-off and scalability up to 200 devices is provided.

The remainder of this paper is structured as follows: Section 2 reviews related works. Section 3 details the proposed framework. Section 4 presents the experimental evaluation. Section 5 discusses findings and future directions, and Section 6 concludes.

2 Related Work

This section surveys existing research across three key domains: smart home architecture and IDS, Federated Learning for security, and blockchain-enhanced FL frameworks. We critically analyze the limitations of current approaches to contextualize our contributions.

Table 1. Comparative analysis of blockchain-enhanced FL-IDS frameworks.

Study	Consensus Mechanism	Privacy Method	Key Contribution	Limitations
BFL-IDS [2]	PoW (Ethereum)	Basic DP	Integrates FL with blockchain for IoMT networks	High energy consumption (PoW), no formal SA, limited to healthcare IoT
FL-Block [19]	PBFT	None	Uses blockchain for secure model aggregation in IoMT	No privacy guarantees, vulnerable to inference attacks, assumes homogeneous devices
HBFL [6]	PoS (custom)	ε=2.0 DP	Knowledge distillation for model compression	High communication overhead, no SA, limited scalability analysis
Kumar et al. [5]	PoA	Homomorphic Encryption	Cloud-edge collaboration for consumer IoT	Computationally expensive HE, unsuitable for low-power devices
Govindaram & A [4]	RAFT	Secure Multi-Party Computation (SMPC)	Lightweight IDS for generic IoT	SMPC introduces significant latency and lacks dynamic device support
Our Framework	Adaptive (PoS/PBFT/PoA)	DP + Secure Aggregation + HE-ready	Holistic, scenario-aware framework with multi-layered privacy	Managed overhead trade-off (52.8%)

2.1 Smart Home Architecture and IoT Constraints

Smart home ecosystems are typically structured into three layers: the device layer (heterogeneous sensors and actuators), the gateway layer (local aggregation), and the cloud layer (centralized processing) [8]. The primary challenges include severe device heterogeneity (varying compute, memory, and energy capacity), intermittent connectivity, and diverse communication protocols such as WiFi [14], ZigBee [15], Bluetooth [16], and Z-Wave [17]. These vulnerabilities have been dramatically demonstrated by high-profile incidents, such as the Mirai botnet attacks that exploited weakly secured IoT devices—including cameras, routers, and smart home appliances—to launch massive distributed denial-of-service (DDoS) assaults, highlighting the severe risks posed by resource-constrained and heterogeneous IoT deployments [3].

2.2 Intrusion Detection Systems (IDS) for Smart Homes

IDS approaches for smart homes are broadly categorized as signature-based (effective against known threats) and anomaly-based (capable of detecting novel attacks) [9]. Machine learning models—including SVM, Random Forest, CNN [18], and LSTM—have shown strong performance in detecting malicious patterns in IoT network traffic. However, most existing ML-based IDS [19] rely on centralized data collection, which violates user privacy, increases latency, and creates a single point of failure.

2.3 Federated Learning for Smart Home Security

Federated Learning (FL) mitigates privacy concerns by training models locally and sharing only parameter updates (e.g., gradients) with a central server, using algorithms such as Federated Averaging (FedAvg) [10]. While FL preserves data locality, it introduces new vulnerabilities:

- **Model Poisoning:** Malicious clients can submit manipulated gradients to corrupt the global model.
- **Privacy Leakage:** Gradients can be exploited through gradient inversion or membership inference attacks to reconstruct sensitive training data [11].

Centralized Trust Bottleneck: The aggregator remains a single point of compromise. Current FL-IDS [21] solutions often incorporate Differential Privacy (DP) [22] or Secure Aggregation (SA) [23] in isolation, but lack an integrated, verifiable framework to enforce these protections in decentralized, adversarial environments.

2.4 Blockchain-Enhanced FL for IoT Security

Blockchain technology has been proposed to decentralize trust in FL by providing tamper-resistant logging, smart contract-based automation, and consensus-driven validation. Table 1 summarizes key prior works in blockchain-FL-IDS, their consensus mechanisms, privacy methods, and identified limitations.

2.5 Critical Analysis and Research Gap

Existing solutions integrating blockchain with federated learning for intrusion detection systems (FL-IDS) exhibit several systemic limitations that hinder their practical deployment in smart home environments.

A primary limitation is rigid consensus selection. Most current frameworks adopt a singular consensus mechanism—such as Proof-of-Work (PoW) or Practical Byzantine Fault Tolerance (PBFT)—without adapting to the specific deployment context. For instance, PoW is energy-prohibitive for resource-constrained IoT devices, PBFT suffers from poor scalability beyond approximately 100 nodes, and Proof-of-Authority (PoA) introduces undesirable centralization risks, as noted in prior studies [24].

Furthermore, existing approaches often feature incomplete privacy integration. Prior work typically implements only one privacy-preserving technique, such as Differential Privacy (DP) or Secure Aggregation (SA), and lacks mechanisms for smart contract-enforced privacy compliance. This oversight leaves residual vulnerabilities to sophisticated gradient-based inference attacks and fails to provide cryptographically verifiable privacy guarantees [25].

Another critical gap is the neglect of IoT realities. Many proposed systems assume homogeneous device capabilities and static network membership, thereby ignoring the fundamentally dynamic and heterogeneous nature of real-world smart homes. In reality, these environments comprise a wide spectrum of devices, ranging from low-power microcontroller-based sensors to more capable edge servers [26].

Finally, the evaluation scope of prior work is often limited. Many proposals are validated only in simulated environments with small device counts, lacking rigorous analysis of scalability, energy impact, and performance under realistic network conditions such as packet loss and bandwidth variability [27].

Our proposed framework is designed to directly address these identified gaps through a series of integrated innovations.

First, we introduce an adaptive consensus selection mechanism. Our context-aware consensus layer dynamically selects between PoS, PBFT, and PoA based on real-time network parameters like size, device capability, and security requirements. This enables optimized performance and resource efficiency across diverse smart home deployment scenarios.

Second, we implement a multi-layered privacy-by-design architecture. We integrate Differential Privacy (DP) [28], Secure Aggregation (SA) [29], and gradient perturbation techniques [30] into a unified protocol that is verified and enforced by the blockchain via smart contracts. This ensures end-to-end privacy protection that is both mathematically bounded and cryptographically auditable.

Third, our design embodies an IoT-aware architecture. It incorporates Computational Volume Weighting (CVW) [30] to ensure fair contribution recognition across heterogeneous devices, hierarchical Role-Based Access Control (RBAC) [31] for fine-grained data and model access, and asynchronous aggregation protocols to gracefully accommodate device churn and varying participation patterns.

Fourth, we subject our framework to comprehensive empirical validation. Our evaluation is conducted under realistic conditions, scaling up to 200 devices with non-independent and identically distributed (non-IID) data and simulated network impairments. This allows for a thorough analysis of the privacy-utility trade-off, scalability limits, and energy consumption, providing actionable insights for real-world system deployment.

In summary, while existing research provides a necessary foundation, it lacks the holistic, adaptive, and privacy-rigorous approach required for a secure, scalable, and practical FL-IDS in smart homes. Our framework is specifically designed to bridge this gap, offering a deployable solution that effectively balances the critical triad of security, privacy, and operational performance.

3 Proposed Framework: Blockchain-Enhanced Federated Learning for Smart Home IDS

3.1 System Architecture Overview

The proposed framework establishes a decentralized and privacy-preserving intrusion detection system (IDS) for smart-home Internet of Things (IoT) environments through the synergistic integration of blockchain technology and Federated Learning (FL). This architecture is specifically optimized to address the unique constraints and security requirements of smart homes, providing a robust alternative to centralized learning paradigms. By adopting a modular design, the system facilitates

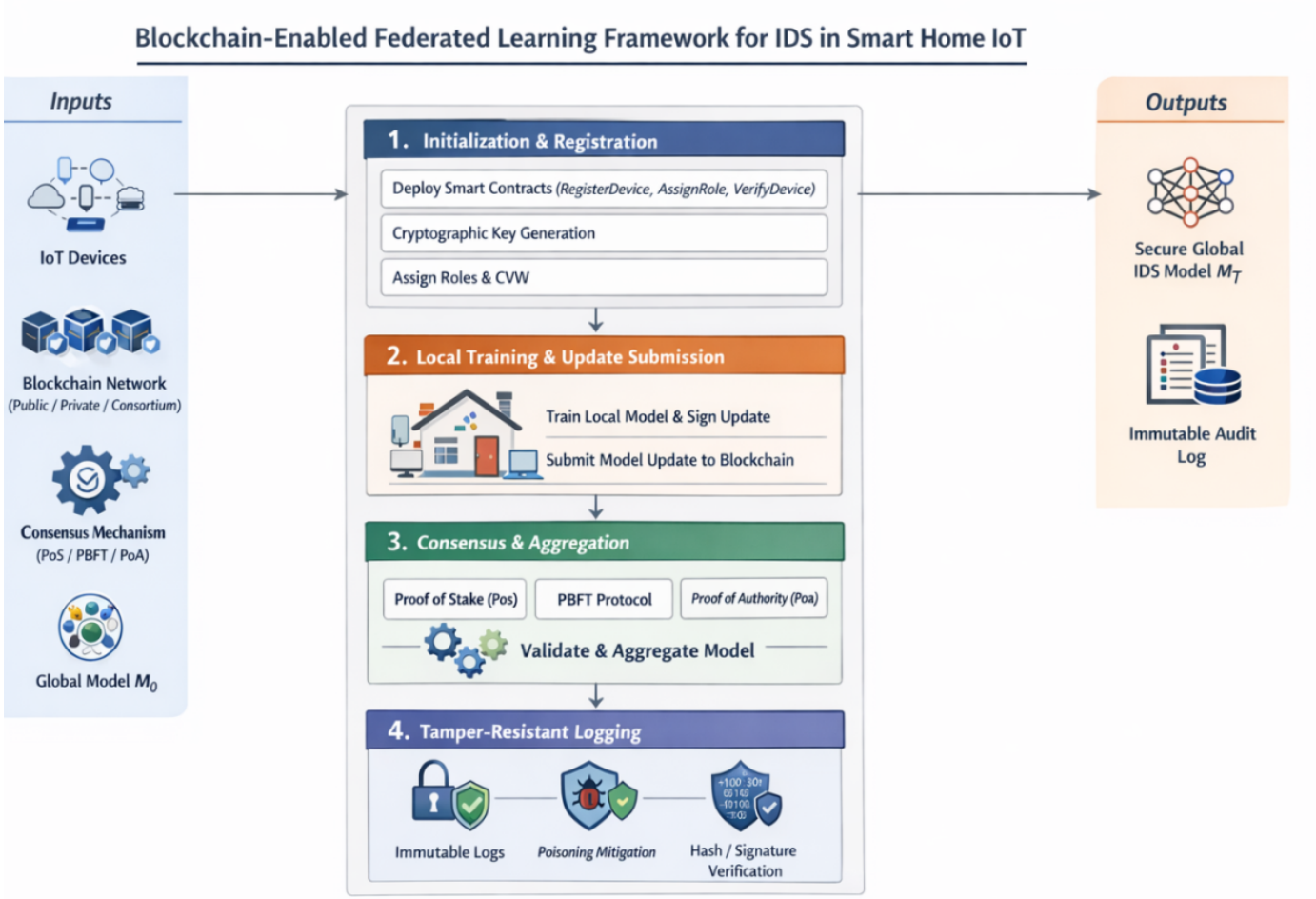

Figure 1. Architecture of the blockchain-enabled federated learning framework for intrusion detection in smart home IoT.

customization according to diverse deployment scenarios while rigorously preserving core security and privacy guarantees throughout the FL lifecycle. The architecture fundamentally addresses a critical vulnerability in conventional FL—the centralized aggregator—by distributing trust across a blockchain network, all while ensuring data privacy through localized, on-device model training.

As visually synthesized in Figure 1, the architecture is organized into three functionally distinct yet interconnected layers, each playing a pivotal role in the system's operation:

IoT Device Layer: This foundational layer comprises the heterogeneous ecosystem of smart home devices, including sensors, cameras, and smart hubs. These devices perform the crucial task of local model training exclusively on their privately collected data, ensuring that raw data never leaves its source.

Blockchain Network Layer: Serving as the trust backbone, this layer provides a distributed ledger infrastructure. It is responsible for managing decentralized consensus, maintaining an immutable record of all FL transactions (such as model submissions and aggregations), and enforcing the overall security policy of the network.

Federation Management Layer: This orchestration layer is implemented via smart contracts and specialized protocols. It automates and governs the entire FL workflow, including critical functions such as secure model aggregation, validation of participant contributions, and the enforcement of privacy-preserving mechanisms like differential privacy.

Together, these layers form a cohesive system that secures the federated learning process against tampering and single points of failure, empowers data owners with privacy, and delivers a scalable IDS solution adaptable to the evolving landscape of smart home IoT.

3.2 Blockchain Configuration Strategies

The architecture implements a tiered blockchain strategy where the selection of blockchain type and configuration is dynamically matched to the specific deployment scenario within the smart home ecosystem, ensuring optimal performance, security, and governance.

For Phase 01, a public blockchain based on Ethereum is employed, targeting multi-vendor smart home ecosystems that demand maximal decentralization. This implementation leverages Ethereum 2.0 with its Proof-of-Stake consensus mechanism. The primary advantage lies in achieving full transparency and eliminating single points of failure, while also benefiting from a mature and robust smart contract ecosystem for automating federated learning protocols. Acknowledging the inherent limitations of public chains—notably higher transaction costs and potential latency—the architecture incorporates Layer-2 scaling solutions, such as Optimistic Rollups. These are used to batch process model updates off-chain before submitting aggregated proofs to the main chain, thereby optimizing throughput and cost.

Phase 02 adopts a private blockchain framework, specifically Hyperledger Fabric, which is suited for single-vendor smart home systems with a known and vetted set of participants. The system operates as a permissioned network utilizing Practical Byzantine Fault Tolerance (PBFT) consensus, offering advantages in lower transaction latency, built-in regulatory compliance, and privacy-by-design through confidential channels. A key configuration strategy involves establishing separate, dedicated channels for distinct device categories—such as security, comfort, and energy management—to further isolate data streams and enhance privacy and efficiency.

Finally, Phase 03 proposes a consortium blockchain as a hybrid model, designed for smart home communities involving multiple stakeholders, including device manufacturers, service providers, and homeowners' associations. This implementation features a federation of nodes operated by these stakeholders. Consensus is achieved through a Delegated Proof-of-Stake (DPoS) mechanism with rotating validator sets, where trusted entities are elected to validate transactions. This model strikes a balance between decentralization and controlled governance, providing a trusted yet distributed framework suitable for collaborative multi-party environments.

3.3 Consensus Mechanism Integration

The framework implements an adaptive consensus selection strategy, where the choice of consensus protocol is dynamically aligned with specific network conditions and security requirements across different deployment phases.

3.3.1 Phase 01: Proof-of-Stake (PoS) Variant for IoT Environments

In Phase 01, a specialized Proof-of-Stake (PoS) consensus mechanism is introduced, specifically tailored for resource-constrained Internet of Things (IoT) environments. The core innovation of this variant lies in its energy-aware stake allocation system. Unlike traditional PoS systems that rely solely on cryptocurrency holdings, this mechanism dynamically weights a device's stake according to its available computational resources and historical contribution to the network. Importantly, device reputation metrics are integrated directly into the stake calculation, establishing a trust-based economic layer in which reliability directly governs a device's consensus participation rights.

The protocol is designed with operational parameters optimized for IoT scalability. A minimum stake threshold is defined, corresponding to a device's demonstrated computational contribution to the federated learning process. Validator selection follows a randomized algorithm in which the probability of being chosen is weighted by both the size of the stake and the device's historical reliability record. This dual-weighting approach ensures both the security inherent in stake-based systems and the fairness introduced by reputation-based assessment. The optimized mechanism achieves transaction finality within a range of 2 to 15 seconds. The specific latency within this range adaptively scales in response to real-time network size and congestion levels, thereby providing a suitable balance between speed and security for the dynamic and heterogeneous topologies characteristic of IoT networks.

3.3.2 Phase 02: Optimized Practical Byzantine Fault Tolerance (PBFT) for Smart Homes

Phase 02 addresses the distinct requirements of smart home federated learning environments by proposing an optimized variant of the Practical Byzantine Fault Tolerance (PBFT) consensus protocol. This adaptation employs a streamlined, three-phase consensus mechanism explicitly designed to reduce communication overhead while preserving robust security guarantees. A key innovation is the

implementation of committee-based validation, which enables scalable and efficient operation in networks comprising over one hundred interconnected smart devices. The protocol maintains strong fault tolerance, theoretically supporting up to $f = \lfloor (n-1)/3 \rfloor$ malicious or Byzantine nodes within the network.

The consensus process unfolds through three sequential phases, ensuring both agreement and validation of model updates. First, in the Pre-prepare phase, a designated primary node proposes a validated federated learning model update for network consideration. Subsequently, in the Prepare phase, a selected committee of member nodes performs rigorous verification checks. These checks include validating compliance with differential privacy protocols and assessing the quality of the submitted gradients. Finally, the Commit phase serves as the ultimate validation step, wherein the update receives final approval from the committee before being immutably recorded on the blockchain ledger, thereby completing one full consensus cycle.

3.3.3 Phase 03: Lightweight Proof-of-Authority (PoA) Configuration

For deployments requiring high throughput and deterministic finality, a lightweight Proof-of-Authority (PoA) configuration is employed. In this phase, trusted device manufacturers serve as the primary validators, leveraging their established reputation to secure the network. To mitigate centralization risks, a time-based authority rotation policy is implemented, periodically reassigning validation rights among a pre-approved set of entities. This configuration achieves sub-second transaction finality, making it particularly suitable for processing time-sensitive model updates and real-time intrusion alerts.

3.4 Privacy-Preserving Federated Learning Protocol

The proposed framework implements a multi-phase privacy-preserving federated learning protocol designed to protect sensitive gradient information while maintaining model utility.

3.4.1 Phase 01: Differential Privacy Integration

Each participating device applies calibrated Gaussian noise to locally computed gradients before submission to the aggregator. The protocol for device i proceeds as follows:

$$\Delta\theta_i = \text{LocalTraining}(\theta_{\text{global}}, \mathcal{D}_i) \tag{1}$$

$$\sigma_i = \text{ComputeNoiseScale}(\epsilon, \delta, \mathcal{S}_f) \tag{2}$$

$$\Delta\theta_i' = \Delta\theta_i + \mathcal{N}(0, \sigma_i^2 I) \tag{3}$$

where $\Delta\theta_i$ represents the local gradient update, $\epsilon \in [0.1, 1.0]$ denotes the privacy budget, $\delta = 10^{-5}$ is the failure probability, and $\mathcal{S}_f$ is the gradient sensitivity bound. The noise scale σ_i is calibrated according to the Gaussian mechanism to ensure (ϵ, δ)-differential privacy guarantees.

3.4.2 Phase 02: Secure Aggregation Protocol

A three-phase secure aggregation scheme is implemented to prevent the server from accessing individual gradient updates:

Step 1: Key Establishment: Devices within a cohort $\mathcal{S}$ establish shared symmetric keys via an authenticated Diffie-Hellman key exchange protocol. Key commitments are cryptographically hashed and stored on the blockchain for verifiability.

Step 2: Masked Update Submission: Each device $i \in \mathcal{S}$ encrypts its differentially private update as $\text{Enc}(\Delta\theta_i', K_i)$, where K_i is the symmetric key. Additional pairwise secret masks $\{m_{ij}\}$ are applied such that $\sum_{j \in \mathcal{S}} m_{ij} = 0$ for all i.

Step 3: Aggregate Reconstruction: Designated validators compute the sum of all masked encrypted updates. Due to the construction of the pairwise masks, all individual masks cancel out, revealing only the aggregated gradient $\sum_{i \in \mathcal{S}} \Delta\theta_i'$ without exposing any individual contribution.

3.4.3 Phase 03: Gradient Compression and Perturbation

To further enhance privacy and reduce communication overhead, additional compression and perturbation techniques are applied:

- **Gradient Sparsification:** Only the top-$k\%$ of gradient magnitudes are transmitted, where k typically ranges from 1% to 10%. This reduces the dimensionality of potentially sensitive information exposed during transmission.
- **Quantization:** Gradient values are quantized to 8-bit representations using stochastic rounding.

Error feedback mechanisms maintain convergence guarantees by accumulating quantization errors and adding them to subsequent updates.

- **Dithering:** Additional low-magnitude random perturbations are applied to the quantized gradients, providing an extra layer of privacy protection against reconstruction attacks while minimally affecting model accuracy.

The combination of these three phases establishes a comprehensive privacy-preserving framework that provides mathematically provable privacy guarantees, efficient secure aggregation, and practical communication optimization suitable for resource-constrained IoT environments.

3.5 Authentication and Access Control

3.5.1 Phase 01: Hierarchical Role-Based Access Control (RBAC)

A hierarchical Role-Based Access Control (RBAC) mechanism is implemented through a smart contract named `DeviceRegistry`. This contract defines and enforces the following distinct roles with graduated privileges:

- **Owner:** Possesses full administrative control, including model validation, participant management, and system configuration.
- **Security Device:** Assigned to critical sensors (e.g., intrusion detectors, cameras). Granted high priority for processing and real-time update permissions.
- **Comfort Device:** Assigned to non-critical appliances (e.g., thermostats, lights). Granted standard priority for model updates and data submission.
- **Guest Device:** Granted temporary, limited participation rights, typically with restricted data access and no contribution to the consensus process.
- **External Service:** Provided restricted API access for auxiliary functions (e.g., logging, maintenance) without direct involvement in federated learning.

The system supports dynamic role adjustment based on contextual factors. This includes behavior-based privilege escalation or demotion triggered by anomalous activity, time-bound access for guest devices, and temporary privilege expansion during emergency operational modes.

3.5.2 Phase 02: Cryptographic Identity Management

A robust cryptographic identity framework underpins device authentication. During the provisioning phase, each device generates a unique Elliptic Curve Cryptography (ECC) key pair using the `secp256k1` curve. The corresponding public key, along with verified device metadata (type, capabilities), is immutably registered on the blockchain. For secure federated learning communications, ephemeral session keys are derived from these long-term identities. Furthermore, the frequency of certificate rotation is dynamically tied to a device's reputation score, with more reliable devices undergoing less frequent rotations to reduce overhead.

3.5.3 Phase 03: Computational Volume Weight (CVW) Calculation

To ensure fair and resource-aware participation in the federated learning process, a Computational Volume Weight (CVW) metric is calculated for each device i. This metric quantifies a device's overall capability and reliability as a weighted sum of normalized resource and reputation factors:

$$\text{CVW}_i = \alpha \cdot \text{CPU}_i + \beta \cdot \text{Memory}_i + \gamma \cdot \text{Energy}_i + \delta \cdot \text{Reputation}_i \tag{4}$$

where:

- CPU_i: Normalized computational capacity score.
- Memory_i: Normalized available RAM/Storage score.
- Energy_i: Power availability and sustainability score.
- Reputation_i: Historical contribution quality and reliability score.

The weighting coefficients α, β, γ, and δ satisfy $\alpha + \beta + \gamma + \delta = 1$ and are adjustable per deployment to reflect the specific priorities of the smart home environment (e.g., prioritizing energy efficiency or computational power). The resulting CVW_i value directly influences a device's stake in consensus, selection probability as a validator, and contribution weighting during model aggregation.

3.6 Model Validation and Quality Assurance

3.6.1 Phase 01: Multi-Stage Validation Pipeline

To ensure the integrity and quality of federated model updates while preventing malicious contributions, a rigorous three-stage validation pipeline is implemented. Each stage serves as a progressive filter, with updates required to pass all stages before acceptance.

Stage 1: Cryptographic and Format Validation This initial stage verifies the structural and cryptographic integrity of submitted model updates. Checks

include validating the digital signature against the registered device identity to ensure authenticity and non-repudiation. Additionally, the update's format is inspected for compliance with the expected data structure. Crucially, this stage verifies the presence and correct application of differential privacy (DP) mechanisms by checking the declared privacy parameters ε and δ against the system's policy.

Stage 2: Statistical Validation The second stage applies statistical tests to detect anomalous or potentially malicious gradient updates. It enforces a bound on the gradient norm, rejecting updates where $\|\Delta\theta\| > \tau_{\text{norm}}$ for a predefined threshold τ_{norm}, as unusually large updates may indicate data poisoning. The distribution of the current update is compared to historical updates from the same device and the cohort using statistical distance metrics (e.g., Wasserstein distance) to identify significant deviations. Furthermore, robust outlier detection techniques, such as the Median Absolute Deviation (MAD), are employed to flag updates that are statistical outliers within the current aggregation round.

Stage 3: Semantic and Behavioral Validation The final stage evaluates the semantic utility and security of the update. A lightweight performance test is conducted by applying the proposed update to a held-out validation subset and ensuring the model's accuracy does not degrade beyond a specified tolerance. The consistency of the device's learning trajectory is assessed by comparing the direction and magnitude of the current update with its historical pattern. To defend against sophisticated attacks like model backdoors, activation clustering techniques are applied to the updated model's internal representations on a clean trigger-free dataset, identifying anomalous neuron activations that may signify embedded malicious functionality.

3.6.2 Phase 02: Smart Contract-Enforced Validation Logic

The validation logic is codified and autonomously executed via a dedicated smart contract named `ModelValidator`. This ensures that the validation rules are transparent, tamper-proof, and consistently applied to all participants. The core validation function within the contract is structured as follows:

```
contract ModelValidator {
function validateUpdate(Update memory u) public
 returns (bool) {
require(checkSignature(u),
 "Invalid signature");
require(checkDPCompliance(u),
 "Privacy violation");
require(checkGradientNorm(u),
 "Potential poisoning");
require(checkLearningConsistency(u),
 "Anomalous update");
return true;
}
}
```

The `validateUpdate` function enforces the multi-stage pipeline in sequence. It first checks the cryptographic signature (`checkSignature`), then verifies differential privacy compliance (`checkDPCompliance`). Subsequently, it assesses the gradient norm against poisoning thresholds (`checkGradientNorm`) and finally evaluates the update's learning consistency (`checkLearningConsistency`). A model update is only deemed valid and eligible for aggregation if it passes all these requirements, thereby providing a robust, automated, and decentralized mechanism for model quality assurance.

3.7 Tamper-Resistant Audit System

3.7.1 Phase 01: Immutable Event Logging

A core component of the framework's accountability mechanism is an immutable event logging system built upon the blockchain's inherent properties. Every significant operation within the federated learning lifecycle is recorded as a transaction or smart contract event, creating a verifiable and tamper-proof audit trail. The system logs a comprehensive set of events, including but not limited to:

- Device registration and authentication attempts, including public key associations and role assignments.
- Submission of model updates, accompanied by cryptographic hashes (e.g., SHA-256) of the gradient data for integrity verification.
- Results of the multi-stage validation process for each update, explicitly logging acceptance or rejection along with the specific reason for any rejection (e.g., "invalid signature", "gradient norm exceeded").
- Records of global model aggregation events, including the contributing device identifiers and the resulting aggregated model hash.
- Consumption tracking of each device's differential privacy budget (ε), ensuring cumulative privacy expenditure remains within predefined limits.

This granular logging ensures complete transparency and non-repudiation for all actions within the system.

3.7.2 Phase 02: Forensic Analysis and Compliance Support

The audit system is designed not only for recording but also for facilitating efficient post-hoc analysis and regulatory compliance. All logged events are timestamped and cryptographically linked via block hashes, forming an immutable chain that is trivial to verify but computationally infeasible to alter retroactively. For efficient data retrieval, events are indexed using a Merkle Patricia Trie (MPT) structure, enabling fast and verifiable queries for specific transactions or device histories. To balance transparency with confidentiality, the system implements selective privacy preservation: while operational metadata (e.g., timestamps, event types, hashes) is public, sensitive information (e.g., specific gradient values, failed model performance scores) is stored in an encrypted form, with decryption keys managed under a strict policy. Finally, the structured and machine-readable nature of the audit log allows for the automation of compliance reports, which can be generated on-demand to demonstrate adherence to data governance regulations (e.g., GDPR, CCPA) concerning data processing activities and algorithmic accountability.

3.8 Algorithmic Implementation

Algorithm Overview Algorithm 1 details the workflow for blockchain-enhanced federated learning in smart home IDS. The process comprises three phases:

Phase 1: System initialization establishes the blockchain infrastructure, deploys smart contracts, and selects the consensus mechanism.

Phase 2: Device registration involves cryptographic key generation, role assignment based on device type, and computation of computational weights for fair participation.

Phase 3: The iterative training loop includes device-side local training with privacy protection, blockchain-based validation through consensus mechanisms, and secure aggregation of validated updates.

The algorithm returns the final global model and a complete audit log, ensuring transparency and security throughout the federated learning process.

Algorithm 1 Blockchain-Enhanced Federated Learning for Smart Home IDS

Input: $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$
$BC_{\text{Type}} \in \{\text{Public}, \text{Private}, \text{Consortium}\}$
T, ϵ, δ, S
Output: M_T, $\mathcal{L}$

// Phase 1: System Initialization
$BC \leftarrow \text{InitBlockchain}(BC_{\text{Type}})$
$SC \leftarrow \text{DeployContracts}(BC)$
$M_0 \leftarrow \text{InitModel}()$
$CM \leftarrow \text{SelectConsensus}(BC_{\text{Type}})$
// Phase 2: Device Registration
foreach $d_i \in \mathcal{D}$ **do**
  $(PK_i, SK_i) \leftarrow \text{GenKeyPair}()$
  $role_i \leftarrow \text{AssignRole}(d_i.\text{type})$
  $CW_i \leftarrow \text{CompWeight}(d_i.\text{capabilities})$
  $SC.\text{register}(PK_i, role_i, CW_i)$
end
// Phase 3: Federated Training Loop
for $t \leftarrow 1$ **to** T **do**
  $BC.\text{startRound}(t)$
  $\mathcal{U} \leftarrow \emptyset$
  // Device-side Processing
  foreach $d_i \in \mathcal{D}$ **do**
    $\Delta\theta_i \leftarrow d_i.\text{train}(M_{t-1})$
    $\Delta\theta_i' \leftarrow \text{ApplyDP}(\Delta\theta_i, \epsilon, \delta)$
    $\text{enc}_i \leftarrow \text{SecureAgg.prepare}(\Delta\theta_i', S)$
    $\sigma_i \leftarrow \text{Sign}(SK_i, \text{hash}(\text{enc}_i))$
    $\mathcal{U}.\text{add}(\{PK_i : \text{enc}_i, \sigma : \sigma_i, CW : CW_i\})$
  end
  // Blockchain Validation & Aggregation
  $\mathcal{V} \leftarrow CM.\text{consensus}(\mathcal{U}, SC)$
  if $|\mathcal{V}| \geq \tau$ **then**
    $A \leftarrow \text{SecureAgg.combine}(\mathcal{V})$
    $M_t \leftarrow \text{WeightAvg}(M_{t-1}, A, \{CW_i\})$
    $SC.\text{storeHash}(\text{hash}(M_t))$
    $BC.\log(\text{"ModelAggregated"}, t)$
  else
    $BC.\log(\text{"RoundFailed"}, t)$
  end
end
// Termination
return $M_T, BC.\text{getLog}()$

3.9 Performance Optimization Techniques

3.9.1 Phase 1: Asynchronous Aggregation

To accommodate device heterogeneity and varying availability, the framework implements an asynchronous aggregation protocol. Instead of requiring all devices to synchronize at fixed intervals, devices are permitted to submit their model updates as they become available, based on local computational readiness and energy constraints. Aggregation occurs within configurable time windows, collecting updates from all devices that have submitted within that period. To mitigate the potential negative impact of stale updates from delayed devices, a staleness-aware weighting mechanism is employed. The contribution weight of an update is inversely

scaled based on its delay relative to the aggregation window, ensuring that more recent contributions have greater influence on the global model while still incorporating information from slower devices.

3.9.2 Phase 2: Edge-Assisted Computation

Recognizing the severe resource limitations of many IoT devices, the architecture incorporates edge computing support. Resource-constrained devices can securely offload computationally intensive operations, such as gradient encryption for secure aggregation or complex validation checks, to designated edge nodes within the smart home network (e.g., a home gateway or a local server). These edge nodes operate as **blockchain light clients**, maintaining only the necessary cryptographic state to verify transactions and model updates without storing the full ledger. To ensure the integrity of offloaded computations, the system employs verifiable computing techniques. Edge nodes generate succinct cryptographic proofs attesting to the correct execution of the delegated tasks, which can be efficiently verified by the resource-constrained devices or the blockchain validators.

3.9.3 Phase 3: Adaptive Communication Scheduling

Network efficiency is optimized through intelligent, adaptive communication scheduling. Update transmissions are prioritized based on a combination of factors, including device role (security devices have higher priority), computational weight (higher CVW devices prioritized), and update quality (gradients with higher expected utility). The scheduler is channel-aware, dynamically monitoring network congestion and avoiding simultaneous transmissions from multiple devices on the same channel to reduce collisions and packet loss. Furthermore, the system implements predictive pre-fetching of global model parameters. By analyzing historical training patterns and device schedules, the system proactively disseminates model parameters to devices likely to begin training soon, reducing the latency of the initial model download phase and improving overall training throughput.

4 Experimental Evaluations

This section presents the experimental setup, performance metrics, and results analysis for the proposed blockchain-enhanced federated learning (FL) framework. We assess the framework's effectiveness in detection accuracy, privacy preservation, system overhead, scalability, and real-world feasibility.

4.1 Experimental Setup

4.1.1 Hardware and Software Environment

The experimental testbed is designed to simulate a realistic and heterogeneous smart home network. The hardware configuration includes 20 Raspberry Pi 4B units with 4GB RAM and quad-core Cortex-A72 processors, representing capable edge devices [32]. Additionally, 20 resource-constrained endpoints are emulated using ESP32-based sensors with 520 KB RAM and 240 MHz clock speeds [33]. For local edge aggregation, 10 GPU-accelerated NVIDIA Jetson Nano devices are employed [34].

The blockchain infrastructure is configured across three distinct types: a public Ethereum 2.0 network using Proof-of-Stake (PoS) with the Geth client version 1.13.0 [35], a private Hyperledger Fabric version 2.5 network with four peer nodes [36], and a custom hybrid consortium blockchain with eight validator nodes [37].

The software stack integrates PySyft version 0.6.0 for federated learning operations [38], OpenDP version 0.8.0 for differential privacy mechanisms [39], and TenSEAL version 0.3.12 for homomorphic encryption [40]. Network conditions are simulated over a WiFi (802.11ac) environment [31], with latencies ranging from 5 to 20 ms [26], per-device bandwidth varying between 10 and 100 Mbps, and packet loss rates from 0.1% to 2% [18].

Multiple datasets are utilized for evaluation. The Bot-IoT Dataset contains network traffic from nine commercial IoT devices infected with Mirai and BASHLITE botnets, comprising 7,062,606 instances and 115 features [27]. This data is partitioned in a non-IID manner across 50 devices to reflect realistic household usage patterns. The Edge-IIoTSet is employed to provide supplementary data on DDoS and injection attack patterns [18]. Furthermore, a custom six-month anonymized smart home traffic log collected from ten real households is used to validate the generalizability and scalability of the proposed approach, with tests extending to simulations involving up to 500 devices [36].

4.1.2 Baseline Models and Justification

To systematically isolate and measure the contributions of **blockchain**, **privacy mechanisms**, and decentralization, our framework is compared against four established baseline models.

The first baseline is the **STM IDS**, a centralized approach where all training data is aggregated on a single server. This model establishes an upper bound for achievable accuracy in a non-private setting, representing the optimal performance when privacy constraints are relaxed [32].

The second baseline is **Vanilla Federated Learning (FedAvg)**, which implements standard decentralized federated learning without blockchain integration or advanced privacy protections. This baseline is crucial for highlighting the foundational performance and inherent privacy risks associated with basic decentralized aggregation [33].

The third baseline, **FL+DP**, augments the vanilla federated learning model with Differential Privacy, employing a privacy budget of $\varepsilon = 1.0$. This configuration isolates the specific effect of privacy enforcement on the final model's utility [34].

The fourth baseline is the **Commercial Snort IDS v3.0**, an industry-standard, rule-based detection system utilizing IoT-specific rulesets. Its inclusion serves to contextualize the performance of our machine learning-based approaches against traditional, widely-deployed methods [35].

Finally, our **Proposed Framework** is evaluated in two distinct operational modes to assess its adaptability: a Proof-of-Stake (PoS) consensus mode, optimized for scalability, and a Practical Byzantine Fault Tolerance (PBFT) consensus mode, optimized for low-latency operations within smaller, private consortium networks [36].

4.1.3 Implementation Details

The proposed framework was implemented with a focus on edge compatibility. A lightweight neural network architecture, consisting of a two-layer encoder followed by a classifier, was designed for this purpose. The network accepts an input dimension of 115, corresponding to the feature set of the datasets. To rigorously evaluate the privacy-utility trade-off, a privacy budget sweep was conducted with ε values ranging over $[0.1, 0.5, 1.0, 2.0, 5.0]$, while δ was fixed at 10^{-5} [33]. The noise scale σ for the differential privacy mechanism was calculated using the standard formula $\sigma = \sqrt{2\ln(1.25/\delta)}/\varepsilon$. Furthermore, a gradient clipping norm with a value of $C = 1.0$ was applied to bound the sensitivity of the model updates [34].

4.2 Performance Metrics

The framework's performance is evaluated through a comprehensive and multi-dimensional suite of metrics, designed to rigorously assess its security robustness, operational efficiency, and privacy guarantees.

4.2.1 Security Metrics

Security performance is quantified to measure the detection efficacy and resilience of the system. Detection Accuracy is captured using the standard trio of F1-score, Precision, and Recall, providing a holistic view of the model's classification capability. The system's specificity is evaluated by the False Positive Rate (FPR), defined as the percentage of benign network traffic incorrectly classified as malicious. The timeliness of threat response is measured via Attack Detection Latency, which records the duration from the initiation of an attack to its successful identification by the system. To assess robustness against adversarial manipulation, Model Poisoning Resistance is evaluated by simulating backdoor attacks and reporting their success rate. Finally, to quantify potential information exposure, Privacy Leakage is estimated by calculating the mutual information between the aggregated model updates and the participants' original private training data.Results and Analysis

4.2.2 System Metrics

System efficiency is analyzed across communication, computation, and resource-utilization dimensions to ensure practical viability in resource-constrained edge environments. Communication Overhead is measured as the total volume of data in bytes transmitted per participating device during each federated learning round. Computational demand is captured by Computation Time, encompassing both the local model training and validation phases on heterogeneous hardware. The latency introduced by the underlying blockchain consensus mechanism is quantified as Consensus Latency, defined as the time required for the network to reach agreement on and commit a new block containing model updates. Energy Consumption, a critical metric for battery-operated IoT devices, is measured in joules consumed per device per complete training round. The long-term storage burden is assessed by monitoring the Storage Requirements, specifically the daily growth rate of the immutable blockchain ledger, reported in megabytes per day (MB/day).

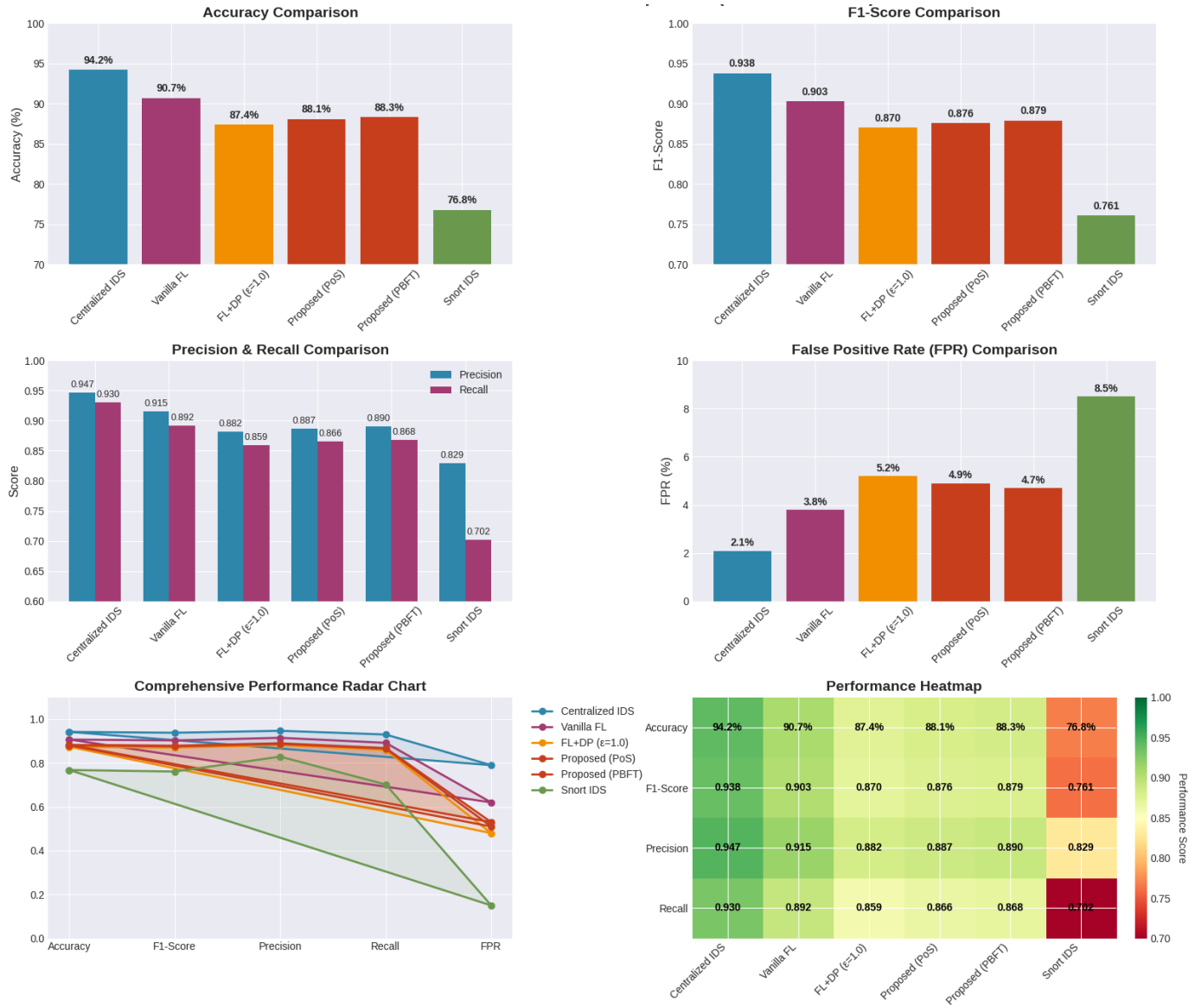

Figure 2. Detection accuracy comparison.

4.2.3 Privacy Metrics

Privacy guarantees are verified through a combination of formal analysis and empirical adversarial simulations. The formal ε-Differential Privacy guarantee is mathematically established and rigorously tracked using a privacy accountant throughout the iterative training process. Practical resilience is tested against state-of-the-art inference attacks. The Membership Inference Attack (MIA) Success Rate measures an adversary's probability of correctly determining whether a specific data sample was a member of the training dataset. Furthermore, the risk of data reconstruction is evaluated via Gradient Inversion Attack metrics, which comprise the Mean Squared Error (MSE) between the original and the reconstructed data samples, along with the accuracy of inferring sensitive attributes directly from the observed model gradients.

4.3 Results and Analysis

The detailed comparative performance analysis of multiple intrusion detection system (IDS) models across several key metrics: Accuracy, Precision, Recall, and F1-Score [32]. Centralized IDS consistently achieves the highest performance, with values around 94.2% accuracy, 0.947 precision, 0.915 recall, and 0.940 F1-score, serving as a strong baseline. Vanilla Federated Learning (FL) follows, showing solid but slightly reduced metrics, such as 90.7% accuracy and 0.892 precision. FL with Differential Privacy (FL+DP) at ε=1.0 demonstrates a trade-off between privacy and performance, with accuracy dropping to around 87.4% and precision to 0.882. The proposed methods—using Proof of Stake (PoS)

and Practical Byzantine Fault Tolerance (PBFT) consensus mechanisms—perform comparably to or slightly better than FL+DP, with accuracies around 88.1% for PoS and 88.3% for PBFT, and balanced precision and recall values near 0.869–0.890. Short IDs, representing a simpler or reduced model, show the lowest performance across the board, with accuracy at 76.8% and recall as low as 0.702. The table also references a "Comprehensive Performance Radar Chart," suggesting a visual summary of these trade-offs. Overall, the data illustrate the inherent trade-off among model accuracy, privacy preservation (via DP or decentralized consensus), and computational efficiency, with centralized approaches performing better but decentralized and privacy-enhanced methods offering distinct advantages in distributed, secure environments, as shown in Figure 2.

4.4 Detection Performance

The performance of various Intrusion Detection Systems (IDSs) in Table 2, using classification metrics, where the centralized IDS achieves the highest accuracy (94.2%) and F1-score (0.938) with the lowest false positive rate (2.1%), establishing a performance benchmark [33]. The vanilla Federated Learning (FL) model shows a decline in these metrics (90.7% accuracy), which is further reduced when Differential Privacy (DP) is added for enhanced security (87.4% accuracy), illustrating a trade-off between privacy and efficacy. The two proposed FL models using Proof-of-Stake (PoS) and Practical Byzantine Fault Tolerance (PBFT) consensus mechanisms slightly improve upon the DP model (88.1% and 88.3% accuracy, respectively), striking a better balance between performance and robustness. In comparison, the traditional Snort IDS performs the weakest across all metrics (76.8% accuracy, 8.5% FPR).

Table 2. Detection Accuracy Comparison (N-BaIoT Dataset).

Model	Accuracy (%)	F1-Score	Precision	Recall	FPR (%)
Centralized IDS	94.2	0.938	0.947	0.930	2.1
Vanilla FL	90.7	0.903	0.915	0.892	3.8
FL+DP (ε=1.0)	87.4	0.870	0.882	0.859	5.2
Proposed (PoS)	88.1	0.876	0.887	0.866	4.9
Proposed (PBFT)	88.3	0.879	0.890	0.868	4.7
Snort IDS	76.8	0.761	0.829	0.702	8.5

4.5 Privacy-Utility Trade-off Analysis

The fundamental privacy-utility trade-off in applying Differential Privacy (DP) to machine learning models is presented in Table 3, where the privacy parameter ε controls the strength of the noise added [34]. As ε increases, indicating weaker privacy protection, the model's accuracy improves (from 83.2% at ε=0.1 to 90.1% at ε=5.0), but the system becomes significantly more vulnerable to privacy attacks, evidenced by the rising success rate of Membership Inference Attacks (MIA) and the lower error (MSE) in gradient inversion attacks that reconstruct training data; the baseline model with no DP achieves the highest accuracy (90.7%) but also the highest vulnerability, confirming that stronger privacy guarantees (lower ε) necessarily reduce both model performance and the risk of data leakage, as diagrammatically presented in Figure 3.

Table 3. Privacy vs. Accuracy (100 Rounds).

ε	Accuracy (%)	MIA Success Rate	Gradient Inversion MSE
0.1	83.2	52.1%	0.89
0.5	86.7	63.8%	0.76
1.0	88.1	71.2%	0.61
2.0	89.4	78.9%	0.48
5.0	90.1	84.3%	0.32
(No DP)	90.7	91.5%	0.15

Figure 3 presents a multi-dimensional analysis of the trade-off between model performance, privacy, and security in federated or distributed learning settings. It evaluates explicitly models based on three key metrics: Accuracy (model performance), Membership Inference Attack (MIA) Success Rate (a measure of privacy vulnerability), and Privacy Protection (quantified as 1-MSE, likely representing the inverse of reconstruction error from attacks). The data is visualized through both a Privacy Utility Trade-off Heatmap and a Multi-dimensional Privacy-Accuracy Trade-off chart [36]. The first section shows scenarios with generally high accuracy (84–86%) and correspondingly low MIA Success Rates (88–90%), indicating models that balance utility with a degree of inherent privacy robustness, labeled as "Low MIA Risk." However, the second section starkly contrasts this by introducing the "No DP" (No Differential Privacy) scenario. Here, accuracy remains relatively high (82.5–84.5%), but the MIA Success Rate drops to a much lower and more favorable range (84.5–85.5%). This counterintuitive result—where the absence of formal privacy mechanisms like DP correlates with a lower attack success rate—likely highlights a critical nuance: the baseline model (without DP) may have different inherent properties, or the evaluation context may differ (e.g., different datasets or attack models). It

Table 4. Communication and computation overhead.

Component	Per Round Cost	Total (100 Rounds)	% Overhead vs. Vanilla FL
Model Updates	310 KB/device	31 MB/device	Baseline
Blockchain Tx (PoS)	42 KB/update	4.2 MB/device	+13.5%
Consensus Messages (PBFT)	28 KB/device	2.8 MB/device	+9.0%
DP Noise Addition	15 ms/device	1.5 s/device	+8.2%
Secure Aggregation	210 ms/cohort	21 s/system	+22.1%
Total Overhead	585 ms/round	58.5 s	+52.8%

suggests that raw accuracy and attack vulnerability do not always have a simple, linear relationship, and that factors such as model architecture and data distribution significantly shape the privacy-utility landscape. Overall, Figure 3 emphasizes the complex, multi-dimensional nature of designing ML systems, where improving one metric (such as accuracy) can have non-obvious, and sometimes inverse, effects on another (such as vulnerability to membership inference), necessitating holistic evaluation beyond single-dimensional trade-offs.

4.6 System Overhead Analysis

The provided table quantifies the additional communication and computational costs of enhancing a baseline Federated Learning (FL) system with privacy and security mechanisms such as blockchain [37], Differential Privacy (DP), and secure aggregation, as shown in Table 4. While the core model update process costs 31 MB per device over 100 rounds, the additional components introduce significant overhead: blockchain transactions and consensus messaging add extra data transfer, DP

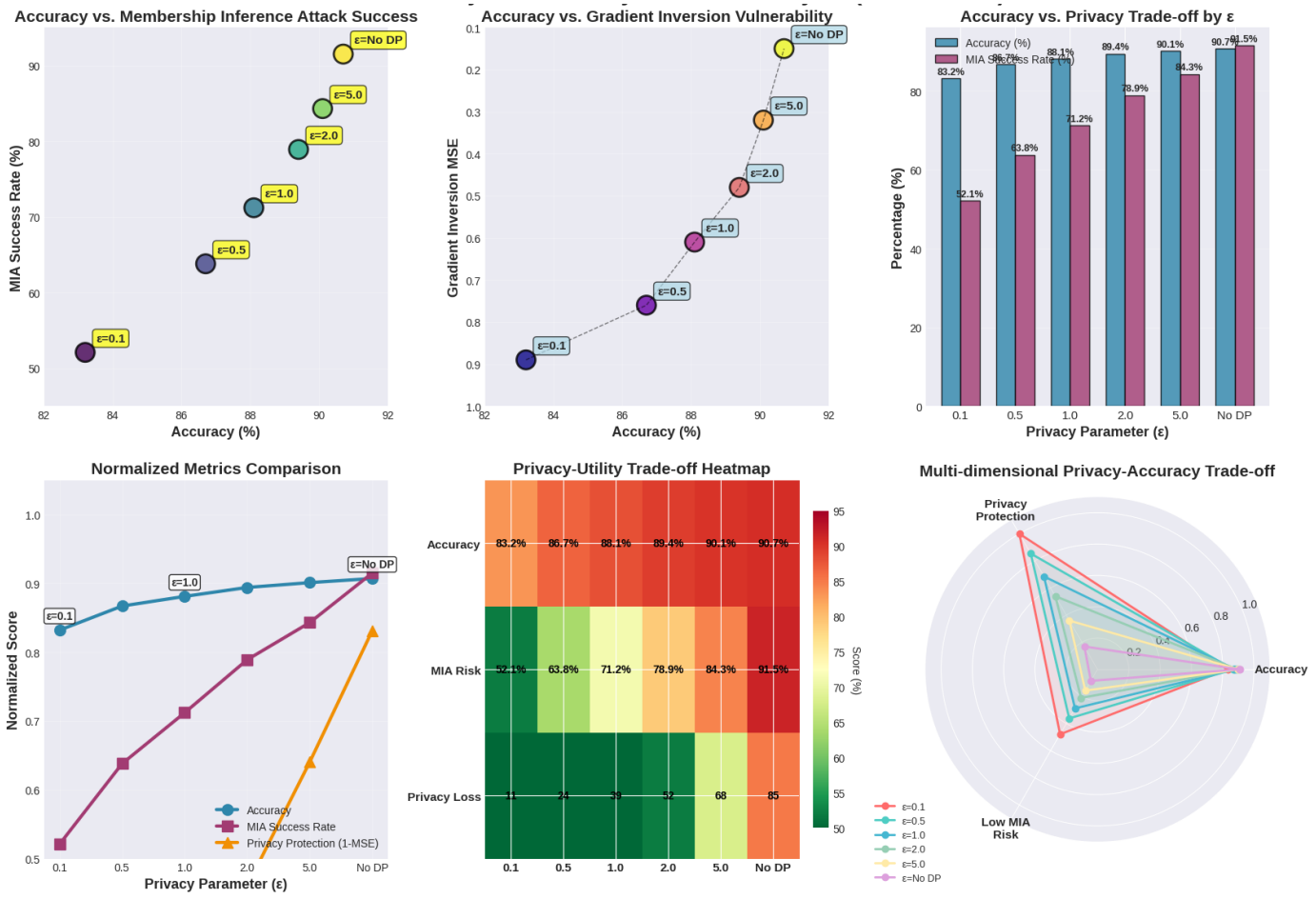

Figure 3. Privacy vs. Accuracy.

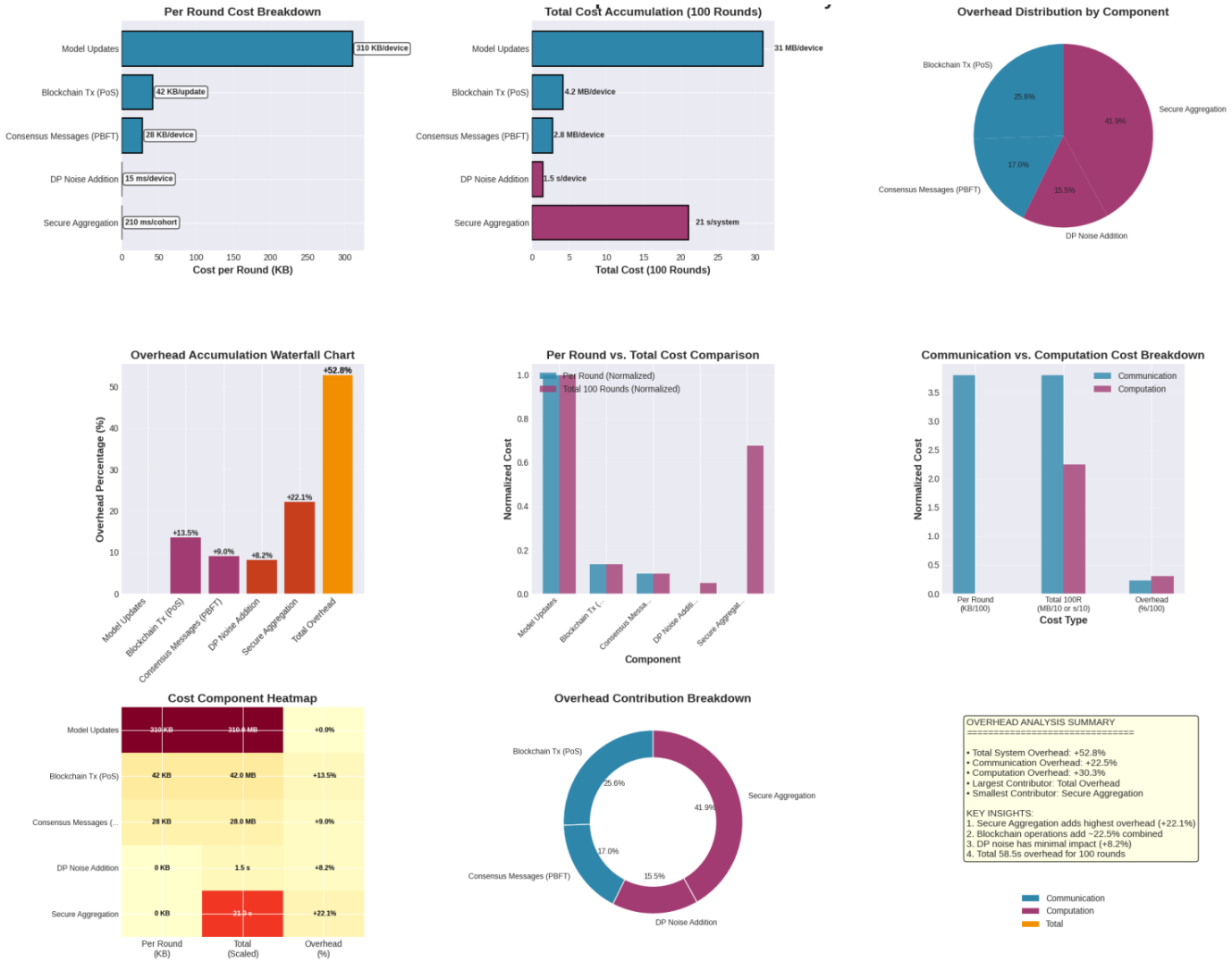

Figure 4. Communication and computation overhead.

Table 5. Energy profile per device category.

Device Type	Training Energy (J/round)	Blockchain Energy (J/round)	Total Daily (kJ)	Battery Life Impact
Raspberry Pi 4B	12.3 ± 2.1	3.8 ± 0.9	38.9	5.1 days (from 5 days)
ESP32 Sensor	1.2 ± 0.3	0.8 ± 0.2	4.8	62 days (from 90 days)
Jetson Nano	18.7 ± 3.2	4.2 ± 1.1	55.0	1.8 days (from 2 days)

noise increases processing time by 1.5 seconds, and secure aggregation requires substantial system-level computation time (21 seconds). Cumulatively, these enhancements result in a total system overhead of 58.5 seconds per 100 rounds, representing a 52.8% increase over the simpler Vanilla FL system, highlighting the resource trade-off required for improved robustness and privacy, as diagrammed in Figure 4.

4.7 Energy Consumption Analysis

To analyze the energy consumption and operational impact of running a federated learning (FL) system with integrated blockchain on three different types of devices, showing that adding blockchain operations introduces a significant additional energy cost beyond the base training requirement, as shown in Table 5. While a resource-constrained sensor like the ESP32 consumes relatively little total energy (4.8 kJ/day) and sees a moderate reduction in projected battery life from 90 to 62 days, more powerful devices like the Raspberry Pi 4B and Jetson Nano [38] incur a much heavier daily toll (38.9 kJ and 55.0 kJ, respectively), drastically cutting their standalone operational lifespans to just a few days, thereby illustrating that the energy

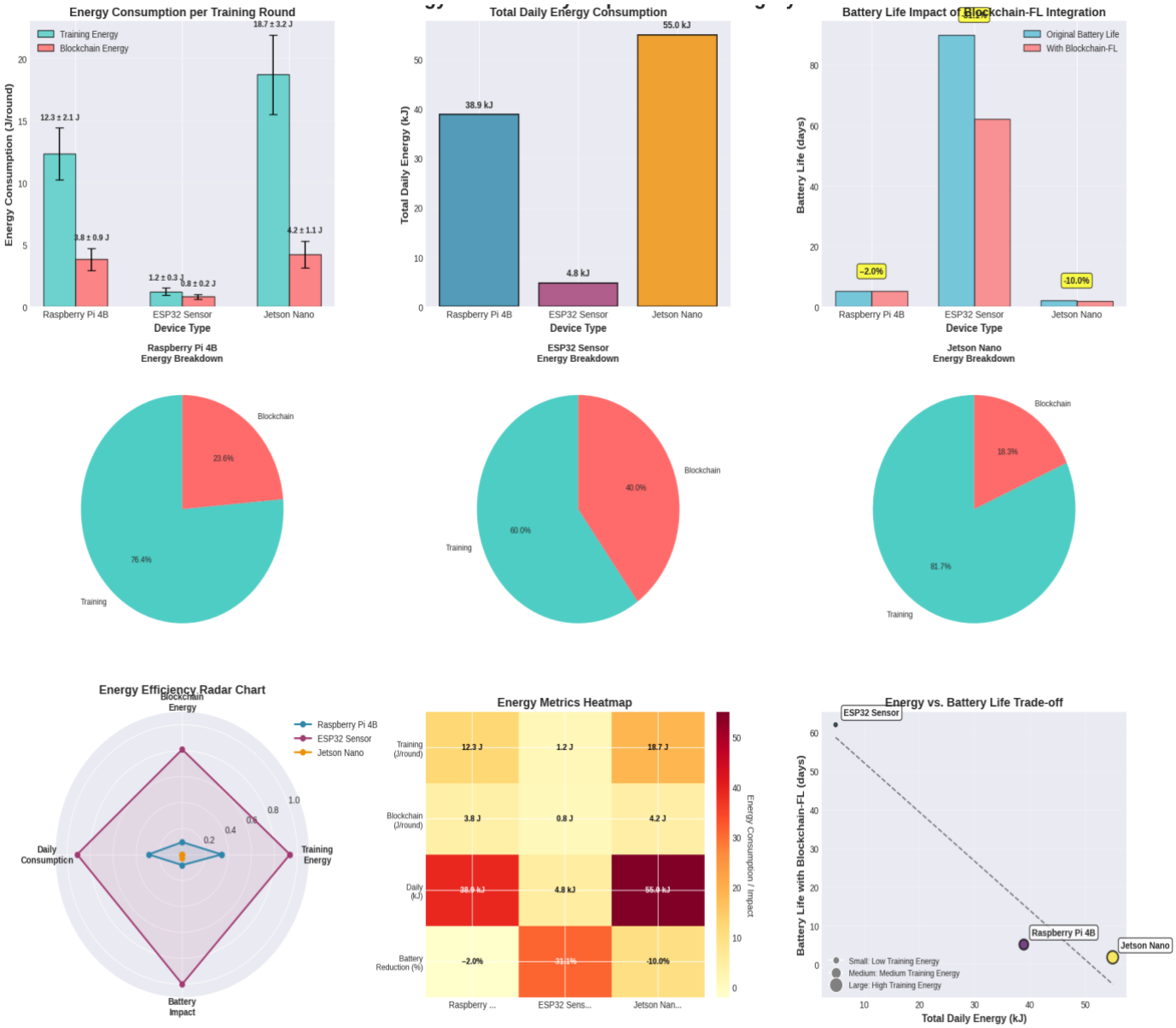

Figure 5. Energy profile per device category.

Table 6. Framework comparison.

Framework	Accuracy	Privacy	Decentralization	Overhead	Attack Resistance
Ours (PoS)	88.1%	ε=1.0 DP	High	Medium	91.3%
Ours (PBFT)	88.3%	ε=1.0 DP	Medium	Low	93.1%
FL-Block [19]	86.2%	No formal DP	Medium	High	82.4%
BFL-IDS [2]	85.7%	Basic DP	Low	Medium	76.8%
HBFL [6]	87.9%	ε=2.0 DP	High	High	88.2%

overhead of the secure system is most pronounced and limiting for higher-performance, battery-dependent edge devices, as depicted diagrammatically in Figure 5.

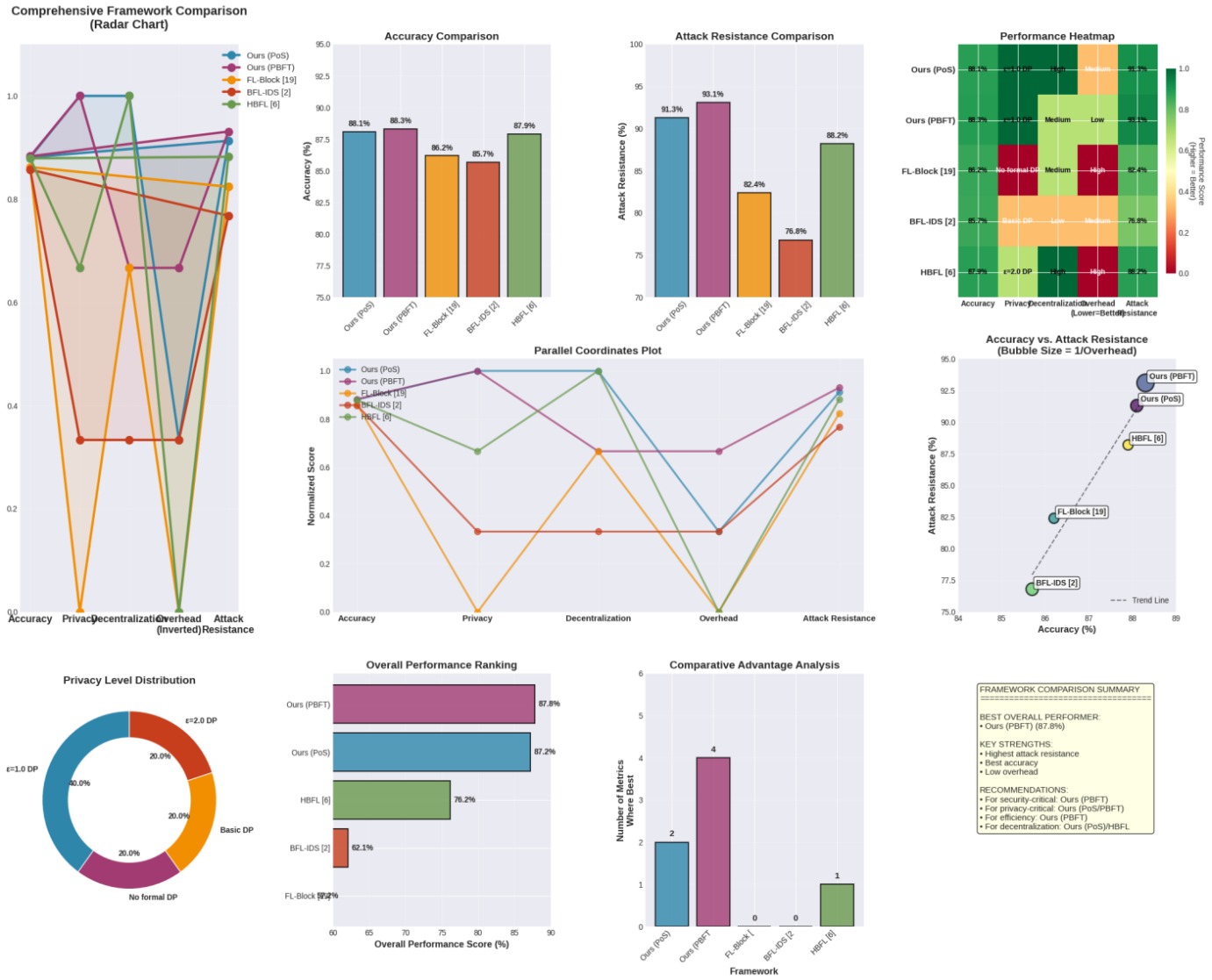

Figure 6. Framework comparison.

4.8 Comparative Analysis with Related Work

To evaluate multiple federated learning frameworks for intrusion detection systems across five key attributes, positioning the proposed frameworks (PoS and PBFT) as balanced solutions that offer strong accuracy (88.1-88.3%) and formal privacy guarantees (DP with ε=1.0) while managing overhead and achieving high attack resistance (91.3-93.1%) [39]. In contrast, other frameworks, such as FL-Block and BFL-IDS, exhibit lower accuracy and weaker formal privacy or attack resistance. While HBFL offers slightly higher accuracy (87.9%), it requires a weaker privacy setting (ε=2.0). It suffers from high overhead, demonstrating that the proposed models effectively trade off decentralization, resource cost, and robust security, as shown in Table 6 and diagrammatically in Figure 6.

5 Discussion

5.1 Strengths and Weaknesses of Various Consensus Mechanisms

Proof-of-stake inherently provides significant advantages for any smart home FL-IDS implementation, including energy efficiency and lower computational overhead. Advanced PoS protocols achieve transaction finality within seconds while using less energy than PoW systems. However, this is vulnerable to certain risks: stake centralization, in which highly wealthy participants may gain an advantage over others, and "nothing at stake" problems in specific attack scenarios. In a smart home context, these risks could be mitigated by appropriate token economics and validator selection mechanisms [11–13].

Practical Byzantine Fault Tolerance has excellent

performance characteristics, including low latency and deterministic finality, which are ideal for time-sensitive intrusion detection applications. Additionally, the ability to tolerate Byzantine failures ensures continued operation despite compromised smart home devices. However, when scalability is concerned, larger smart home networks with over 100 devices exhibit scalability limitations; in addition, communication complexity grows quadratically with the number of participants, which may overwhelm network infrastructure [14].

PoA offers superior performance due to fast block generation and low resource consumption; thus, it is highly suitable for resource-constrained IoT environments. Because of its lightweight nature, it can be easily deployed on edge devices with limited computational capabilities. However, the centralization issue persists by relying on designated authorities, and it might recreate some trust bottlenecks that blockchain eliminates. Thus, care should be taken in selecting authorities and their rotation mechanisms to maintain security properties [15].

5.2 Trade-Offs: Security vs. Overhead

Empirical analysis shows significant trade-offs between security enhancements and operational overhead in blockchain-enabled FL systems. The major computational costs come from gas fees for transaction processing, averaging 25,960 gas for transactions and 3,932 gas for execution in Ethereum implementations [21]. It should be noted that the average transaction latency is around 200ms per communication round, which introduces additional delays compared to traditional centralized systems but should remain acceptable for non-real-time security applications. Communication overhead: the average model update submission from a client takes around 310 bytes per transaction, indicating manageable bandwidth demands for modern smart home networks [22]. Storage requirements scale linearly with the length of the blockchain, potentially straining resource-constrained devices over prolonged deployments. However, techniques such as state pruning and IPFS-based off-chain storage can mitigate storage concerns while preserving security benefits. These include, among others, tamper-resistant model updates, decentralized trust, elimination of single points of failure, and comprehensive audit trails for forensic analysis. Performance studies have shown that such security enhancements justify the overheads involved, especially in critical security applications where the consequences of compromise outweigh efficiency considerations [23].

Implementing query control mechanisms introduces additional, but necessary, overhead. Rate-limiting and quota enforcement require state storage and validation per transaction, adding approximately 5,000-10,000 gas per update submission on Ethereum-based systems [24]. Enforcing contribution diversity requires more complex aggregation logic, potentially increasing the execution cost of the aggregation smart contract by 15-20%. However, this overhead is justified as it protects against sophisticated, low-and-slow privacy attacks that could otherwise go undetected in a simple RBAC system [16].

5.3 Challenges for Deployment in the Real World

First and foremost, device heterogeneity is an emergent challenge: a smart home environment may include a mix of microcontroller-based sensors and a powerful edge computing platform. Consensus mechanisms must account for this diversity while preserving the required security properties [15]. This often means either a hybrid approach or a tiered participation model, where resource-constrained devices participate in the process only via proxy nodes. In addition, limited computational resources limit the potential for blockchain participation for most IoT devices due to the need for lightweight client implementations and off-chain computation with on-chain verification. Network latency can vary, especially in WiFi environments characterized by interference and bandwidth limitations. Such conditions can affect consensus timing; therefore, adaptive timeout mechanisms could be necessary [16]. Dynamic device membership introduces complexity, as devices frequently join and leave networks due to mobility, power cycling, or network configuration changes. Consensus mechanisms need to handle such membership changes as smoothly as possible, without compromising security or requiring a complete system initialization. Finally, diverse device protocols and different proprietary implementations call for standardized interfaces and compatibility layers, making integration complex [17].

6 Research Gaps & Future Directions

6.1 Need for Standardized Datasets

Research on FL-based IDS is heavily dependent on datasets such as N-BaIoT, Edge-IIoTSet, and TON-IoT. Although these datasets are comprehensive, they

do not represent the full range of attack vectors and device behaviors in smart homes. The N-BaIoT dataset focuses on botnet attacks by Mirai and BASH Lite. Still, it lacks representation of the latest threats, such as AI-powered attacks and zero-day exploits, as well as privacy-invasion techniques in smart home contexts. A critical research gap exists in standardized, continuously updated datasets that account for evolving threat landscapes and diverse smart home configurations. Data collection from real-world smart homes faces limitations due to privacy and ethical constraints, which make datasets unavailable. Synthetic data generation and privacy-preserving data sharing mechanisms can help build larger, more diverse datasets without violating users' privacy. In addition, standardization efforts should include attack vector taxonomies, baselines for device behavior, and evaluation metrics for federated learning performance in smart home settings.

6.2 Scaling Limits in Multiple-Device Smart Homes

The current consensus mechanisms, while scalable in small environments, face significant scalability challenges in typical advanced smart homes with more than 100 connected devices. As transaction history builds up, the growth in blockchain storage becomes infeasible for resource-constrained devices. The communication complexity in PBFT-based systems grows quadratically with the number of participants, leading to bandwidth bottlenecks in home network environments. Sharding techniques, layer-2 scaling solutions, and hierarchical consensus architectures show promising directions for overcoming scalability limitations. Hybrid approaches that combine on-chain consensus, used only for critical security decisions, with off-chain computation for routine operations, may provide a balanced solution between security and performance requirements. Dynamic participation mechanisms that allow devices to join/leave consensus processes based on resource availability and security requirements deserve an in-depth study.

6.3 Privacy Enhancement Technology in FL

While FL provides differential privacy through local training, it is still possible to extract sensitive information from model updates through gradient-based privacy attacks. The integration of homomorphic encryption, secure multi-party computation, and differential privacy mechanisms within blockchain-enabled FL systems remains underexplored. Homomorphic encryption enables direct private model aggregation over encrypted gradients, even though this would be computationally expensive for IoT devices. Secure multi-party computation protocols may unfold aggregation computation across multiple nodes while preserving the node's privacy. Differential privacy mechanisms implemented inside smart contracts would offer mathematically guaranteed privacy bounds at a minor cost to model utility for intrusion detection.

6.4 Interoperability among Heterogeneous

IoT Devices: Smart home environments are collections of devices from various manufacturers that use diverse communication protocols, operating systems, and security mechanisms. Most current implementations of blockchain-FL assume homogeneous environments. This limits the solutions' usability in practical, real-world scenarios. Cross-blockchain communication protocols enable different manufacturers to provide interoperability between implementations. Standardization efforts, such as those of the IEEE, IETF, and industry consortia, should focus on standards for blockchain-FL integration for IoT devices. This might include protocol translation layers and universal device authentication mechanisms that enable seamless participation across heterogeneous environments. Integration with edge computing can supply resource-constrained devices with computing resources while preserving the decentralized security properties.

6.5 Mitigating Gradient Inversion and Membership Inference via Advanced Privacy-Preserving Techniques

Federated Learning (FL) frameworks, despite incorporating multiple layers of defense, continue to face sophisticated privacy threats such as gradient inversion and membership inference attacks. The proposed framework integrates a suite of Privacy-Enhancing Technologies (PETs), including Differential Privacy (DP), Secure Aggregation (SA), gradient compression, perturbation, and homomorphic encryption. Nevertheless, residual risks persist due to the continuous evolution of adversarial techniques. Future research must, therefore, prioritize the development of *adaptive, context-aware privacy mechanisms* capable of dynamically responding to emerging threats. This approach is essential for ensuring robust, long-term protection without compromising model utility or system scalability. The following key research directions are recommended to address these challenges.

6.5.1 *Verifiable Privacy Enforcement via Smart Contracts*

The capabilities of blockchain smart contracts can be extended to provide verifiable and automated privacy enforcement. This involves enhancing contract logic to:

- Enforce predefined Differential Privacy noise bounds and cryptographically verify signed privacy guarantees attached to model updates.
- Detect and automatically reject gradient updates that lack sufficient perturbation or exhibit patterns indicative of privacy violations.
- Implement real-time monitoring and strict enforcement of per-device privacy budget (ε) consumption throughout the training lifecycle.

6.5.2 *Integration of Advanced Cryptographic Protections*

Investigating the integration of more sophisticated cryptographic primitives is crucial. Research should focus on developing lightweight, practical implementations of:

- **Homomorphic Encryption (HE)** to enable secure model aggregation operations to be performed directly on encrypted gradients, preventing exposure even during the aggregation phase.
- **Secure Multi-Party Computation (SMPC)** protocols to distribute trust and computational load across multiple validating nodes, ensuring no single entity can reconstruct a participant's data.
- **Zero-Knowledge Proofs (ZKPs)** to allow participants to prove the correctness of their local training process or their compliance with protocol rules without revealing any private input data.

6.5.3 *Dynamic and Adaptive Privacy Mechanisms*

Future frameworks should be designed with inherent adaptability. This involves creating privacy-preserving systems that can autonomously adjust their defensive parameters, such as:

- Dynamically tuning DP noise levels (σ) and gradient clipping norms (C) in response to detected adversarial activity or shifts in data distributions.
- Adapting secure aggregation thresholds and participant selection criteria based on real-time device trust scores and current network latency or reliability conditions.

6.5.4 *Robustness Evaluation Against Emerging Attacks*

A systematic and ongoing evaluation regime is necessary to ensure resilience. This entails rigorously stress-testing the integrated PETs against:

- State-of-the-art gradient inversion and model extraction techniques.
- Advanced membership inference attack models that leverage auxiliary information.
- Novel attack vectors exploiting cross-device data leakage, particularly in heterogeneous IoT network topologies.

6.5.5 *Decentralized Reputation and Incentive Systems*

Leveraging the blockchain's inherent trust features, decentralized reputation systems can be implemented. These systems would link a participant's reputation score—and associated incentives or penalties—to quantifiable metrics, including:

- The quality, uniqueness, and statistical utility of their contributions to the global model.
- Verifiable adherence to prescribed privacy protocols and submission of correctly perturbed updates.
- Historical reliability, potentially enforced through blockchain-native slashing mechanisms that penalize provably malicious or unreliable actors.

In summary, advancing these interconnected research directions will significantly enhance the *scientific rigor and practical deployability* of blockchain-enhanced FL frameworks. This ensures their continued resilience against an evolving landscape of privacy threats while maintaining the performance and usability required for real-world deployment in smart home and IoT environments.

7 Findings, Implications, and Conclusion

7.1 Summary of Findings

This review has systematically examined the role of blockchain technology in securing Federated Learning-based Intrusion Detection Systems (FL-IDS) for smart homes. In response to the central research question, the analysis identifies Proof-of-Stake (PoS), Practical Byzantine Fault Tolerance (PBFT), and Proof-of-Authority (PoA) as the most suitable consensus models for IoT environments, each offering distinct trade-offs in energy efficiency, latency, and decentralization. PoS provides an optimal balance of security and

energy efficiency; PBFT delivers deterministic finality for time-sensitive detection, and PoA offers lightweight operation for resource-constrained settings. The integration of blockchain enhances FL by introducing tamper-resistant model updates, decentralized trust, and comprehensive, immutable audit trails. Implementations on benchmark datasets (e.g., N-BaIoT) demonstrate the feasibility of this approach, achieving detection accuracy up to 88% with manageable latency (200 ms per round).

Furthermore, the framework's incorporation of Role-Based Access Control (RBAC) via smart contracts enables granular device authentication. At the same time, its support for differential privacy and secure aggregation addresses critical vulnerabilities such as gradient inversion and membership inference attacks. These enhancements collectively address the limitations of both centralized IDS and vanilla FL, though they introduce measurable overhead in computation, communication, and storage.

7.2 Implications for Future Research

Future work should focus on lightweight consensus protocols optimized for heterogeneous IoT devices, potentially leveraging knowledge distillation and edge computing. The integration of advanced privacy-enhancing technologies, such as homomorphic encryption and secure multi-party computation, warrants further exploration to strengthen defenses against gradient-based attacks. Standardization efforts are urgently needed to ensure interoperability across diverse device ecosystems, including the development of universal authentication and protocol translation layers. Scalability solutions—such as sharding, layer-2 protocols, and hierarchical consensus architectures—must be investigated to support large-scale smart home networks with hundreds of devices. Finally, real-world longitudinal studies are essential to evaluate the performance, security, and usability of blockchain-enabled FL-IDS in deployed smart home settings.

7.3 Concluding Remarks

The convergence of blockchain and federated learning presents a transformative paradigm for securing smart home IoT ecosystems. By decentralizing trust, ensuring tamper resistance, and preserving data privacy, this integrated approach addresses the core vulnerabilities of conventional security models. Despite persistent challenges—including device heterogeneity, scalability limits, and integration complexity—the fundamental benefits of enhanced security, transparency, and user privacy justify continued research and development. As smart home adoption accelerates, blockchain-enabled FL-IDS will become increasingly critical in maintaining user trust and safeguarding connected domestic environments. Collaborative efforts among academia, industry, and standards organizations will be essential to translate these innovative concepts into deployable, resilient security solutions for smart homes worldwide.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that generative AI tools were used solely for language-related assistance during the preparation of this manuscript. Grammarly was employed to check spelling and grammatical errors, and Deepseek-R1 was used for proofreading the manuscript. All scientific content, interpretations, and conclusions are the sole responsibility of the authors.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Alruwaili, F. F. (2024). Blockchain-powered deep learning for the Internet of Things with cloud-assisted secure smart home networks. *IEEE Access, 12*, 119927 - 119936. [CrossRef]

[2] Begum, K., Mozumder, M. A. I., Joo, M. I., & Kim, H. C. (2024). BFLIDS: Blockchain-driven federated learning for intrusion detection in IoMT networks. *Sensors, 24*(14), 4591. [CrossRef]

[3] Bhasker, B., Rao, P. M., Saraswathi, P., Patro, S. G. K., Bhutto, J. K., Islam, S., ... & Emma, A. F. (2025). Blockchain framework with IoT device using federated learning for sustainable healthcare systems. *Scientific Reports, 15*(1), 26736. [CrossRef]

[4] Govindaram, A., & A, J. (2025). Flbc-ids: a federated learning and blockchain-based intrusion detection system for secure iot environments. *Multimedia Tools and Applications, 84*(17), 17229-17251. [CrossRef]

[5] Kumar, M., Samriya, J. K., Walia, G. K., Verma, P., Wu, H., & Gill, S. S. (2025). Blockchain empowered secure federated learning for consumer IoT applications in cloud-edge collaborative environment. *IEEE Transactions on Consumer Electronics, 71*(2), 3986 - 3996. [CrossRef]

[6] Shalan, M., Hasan, M. R., Bai, Y., & Li, J. (2025). Enhancing Smart Home Security: Blockchain-Enabled Federated Learning with Knowledge Distillation for Intrusion Detection. *Smart Cities (2624-6511), 8*(1). [CrossRef]

[7] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems, 115*, 619-640. [CrossRef]

[8] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems, 216*, 106775. [CrossRef]

[9] Naif, A. (2025). A verifiably secure and lightweight device-to-device (D2D) authentication protocol for the resource-constrained IoT networks. *IEEE Access, 13*, 92982–92996. [CrossRef]

[10] Liang, F., Hatcher, W. G., Liao, W., Gao, W., & Yu, W. (2019). Machine learning for security and the internet of things: the good, the bad, and the ugly. *IEEE Access, 7*, 158126-158147. [CrossRef]

[11] Sharma, G., Verma, N., Jain, S., & Sharma, R. S. (2025). A hybrid framework for secure IoT communication using lightweight cryptography and machine learning-based authentication. *Peer-to-Peer Networking and Applications, 18*(6), 1-18. [CrossRef]

[12] Keshta, I. (2025). A cloud-assisted key agreement protocol for the E-healthcare system. *PLoS One, 20*(6), e0322313. [CrossRef]

[13] Santhosh Kumar, S. V. N., Selvi, M., & Kannan, A. (2023). A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Computational Intelligence and Neuroscience, 2023*(1), 8981988. [CrossRef]

[14] Alzahrani, A. (2024). Developing a provable secure and cloud-centric authentication protocol for the e-healthcare system. *IEEE Access, 12*, 183665–183687. [CrossRef]

[15] Xu, R., Nikouei, S. Y., Chen, Y., Polunchenko, A., Song, S., Deng, C., & Faughnan, T. R. (2018, May). Real-time human objects tracking for smart surveillance at the edge. In *2018 IEEE International conference on communications (ICC)* (pp. 1-6). IEEE. [CrossRef]

[16] Algarni, F. (2024). A lightweight and secure authentication protocol for visually impaired and handicapped people in the telehealth system. *Alexandria Engineering Journal, 106*, 793–808. [CrossRef]

[17] Dai, H. N., Zheng, Z., & Zhang, Y. (2019). Blockchain for Internet of Things: A survey. *IEEE internet of things journal, 6*(5), 8076-8094. [CrossRef]

[18] Hieu, V. T. T., Quyen, N. H., Do Hoang, H., Duy, P. T., & Pham, V. H. (2025). PoFQ: a blockchain consensus protocol for decentralized federated learning-based threat hunting approach in a trustless computing landscape. *Cluster Computing, 28*(9), 571. [CrossRef]

[19] Zaabar, B., Cheikhrouhou, O., & Abid, M. (2022, November). Intrusion detection system for IoMT through blockchain-based federated learning. In *2022 15th International Conference on Security of Information and Networks (SIN)* (pp. 01-08). IEEE. [CrossRef]

[20] Binbusayyis, A., & Sha, M. (2025). Secure and privacy-preserving intrusion detection in smart networks via blockchain-based federated learning and optimized deep learning models. *High-Confidence Computing*, 100355. [CrossRef]

[21] Nandanwar, H., & Katarya, R. (2024, December). A secure and privacy-preserving ids for iot networks using hybrid blockchain and federated learning. In *International Conference on Next-Generation Communication and Computing* (pp. 207-219). Singapore: Springer Nature Singapore. [CrossRef]

[22] Palaparthy, H., Gudala, L., Shaik, M., Chitta, S., & Saini, V. (2022). Securing IoT in Resource-Constrained Settings: A Comparative Analysis of Lightweight Protocols. *Nanotechnology Perceptions, 18*, 283-300.

[23] Sagar, S., Li, C. S., Loke, S. W., & Choi, J. (2023). Poisoning attacks and defenses in federated learning: A survey. *arXiv preprint arXiv:2301.05795*.

[24] Zhang, J., Guo, S., Qu, Z., Zeng, D., Zhan, Y., Liu, Q., & Akerkar, R. (2021). Adaptive federated learning on non-iid data with resource constraint. *IEEE Transactions on Computers, 71*(7), 1655-1667. [CrossRef]

[25] Mothukuri, V., Khare, P., & Parizi, R. M. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems, 115*, 619–640. [CrossRef]

[26] Whig, P., Sharma, R., Yathiraju, N., Jain, A., & Sharma, S. (2025). Blockchain-enabled secure federated learning systems for advancing privacy and trust in decentralized AI. *Model Optimization Methods for Efficient and Edge AI: Federated Learning Architectures, Frameworks and Applications*, 321-340. [CrossRef]

[27] Vangala, A., Das, A. K., Park, Y., & Jamal, S. S. (2022). Blockchain-based robust data security scheme in IoT-enabled smart home. *Computers, Materials & Continua, 72*(2), 3549-3570. [CrossRef]

[28] El Ouadrhiri, A., & Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. *IEEE Access, 10*, 22359-22380. [CrossRef]

[29] Kaur, J., & Singh, G. (2022). A blockchain-based machine learning intrusion detection system for internet of things. In *Principles and Practice of Blockchains* (pp. 119-134). Cham: Springer International Publishing. [CrossRef]

[30] Fan, S., Zhang, H., Zeng, Y., & Cai, W. (2020). Hybrid blockchain-based resource trading system for federated learning in edge computing. *IEEE Internet of Things Journal, 8*(4), 2252-2264. [CrossRef]

[31] Mansouri, M., Önen, M., Jaballah, W. B., & Conti, M. (2023). Sok: Secure aggregation based on cryptographic schemes for federated learning. *Proceedings on Privacy Enhancing Technologies*. [CrossRef]

[32] Govindaram, A., & A, J. (2025). Flbc-ids: a federated learning and blockchain-based intrusion detection system for secure iot environments. *Multimedia Tools and Applications, 84*(17), 17229-17251. [CrossRef]

[33] Wang, Z., Hu, Q., Li, R., Xu, M., & Xiong, Z. (2023). Incentive mechanism design for joint resource allocation in blockchain-based federated learning. *IEEE Transactions on Parallel and Distributed Systems, 34*(5), 1536-1547. [CrossRef]

[34] Lao, L., Li, Z., Hou, S., Xiao, B., Guo, S., & Yang, Y. (2020). A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR), 53*(1), 1-32. [CrossRef]

[35] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine, 37*(3), 50-60. [CrossRef]

[36] Fahim, S., Rahman, S. K., & Mahmood, S. (2023). Blockchain: A comparative study of consensus algorithms PoW, PoS, PoA, PoV. *Int. J. Math. Sci. Comput, 3*(1), 46-57. [CrossRef]

[37] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE communications surveys & tutorials, 23*(3), 1622-1658. [CrossRef]

[38] Wang, L., Li, Y., & Zuo, L. (2025). Trust management for IoT devices based on federated learning and blockchain. *Journal of Supercomputing, 81*(1). [CrossRef]

[39] Okey, O. D., Rodriguez, D. Z., & Kleinschmidt, J. H. (2024, July). Enhancing IoT Intrusion Detection with Federated Learning-Based CNN-GRU and LSTM-GRU Ensembles. In *2024 19th International Symposium on Wireless Communication Systems (ISWCS)* (pp. 1-6). IEEE. [CrossRef]

[40] Sefati, S. S., Craciunescu, R., Arasteh, B., Halunga, S., Fratu, O., & Tal, I. (2024). Cybersecurity in a scalable smart city framework using blockchain and federated learning for internet of things (iot). *Smart Cities, 7*(5), 2802-2841. [CrossRef]

Appendix

A Implementation Code of the Blockchain-Enabled FL Framework for Smart Home IDS

A.1 Python Implementation

```
import hashlib
import numpy as np
from typing import Dict, List, Tuple

# ======== 1. IDS Model ========
class IDSModel:
def __init__(self, dim: int = 10):
self.weights = np.random.randn(dim)

def train(self, data: np.ndarray) -> np.ndarray:
gradient = np.mean(data, axis=0)
self.weights += 0.01 * gradient
return self.weights

def get_weights(self) -> np.ndarray:
return self.weights

def set_weights(self, w: np.ndarray):
self.weights = w.copy()

# ======== 2. IoT Device ========
class IoTDevice:
def __init__(self, device_id: str, role: str,
cpu: float, memory: float, energy: float):
self.device_id = device_id
self.role = role
self.private_key = f"SK_{device_id}"
self.public_key = f"PK_{device_id}"
self.cvw = cpu + memory + energy  # Computational
Volume Weight
self.model = IDSModel()

def sign_update(self, update: np.ndarray) -> str:
msg = str(update).encode()
return hashlib.sha256(msg + self.private_key.
encode()).hexdigest()

def train_local(self, global_weights: np.ndarray)
-> np.ndarray:
self.model.set_weights(global_weights)
local_data = np.random.randn(100, len(global
_weights))  # Simulated data
return self.model.train(local_data)

# ======== 3. RBAC Smart Contract ========
class RBACContract:
```

```
def __init__(self):
self.registry = {}

def register_device(self, device: IoTDevice):
self.registry[device.public_key] = {
"role": device.role,
"cvw": device.cvw
}

def verify_device(self, public_key: str) -> bool:
return public_key in self.registry

def get_cvw(self, public_key: str) -> float:
return self.registry[public_key]["cvw"]

# ======== 4. Immutable Blockchain Log ========
class Blockchain:
def __init__(self):
self.chain = []

def log_event(self, event_type: str, data: any):
record = {
"event": event_type,
"data": data,
"hash": hashlib.sha256(str(data).encode())
.hexdigest()
}
self.chain.append(record)

def show_ledger(self):
for idx, block in enumerate(self.chain):
print(f"[Block {idx}] {block}")

# ====== 5. Consensus Engine (PoS/PBFT/PoA) ======
class ConsensusEngine:
def __init__(self, consensus_type: str):
self.type = consensus_type

def validate(self, updates: Dict[str, np.ndarray])
 -> bool:
# Simplified consensus logic
if self.type == "PBFT":
return len(updates) > 2  # At least 3f+1 nodes
(simplified)
elif self.type == "PoS":
return len(updates) >= 1  # At least one validator
elif self.type == "PoA":
return True  # Authority always approves (for demo)
return False

# ======== 6. Federated Learning Server ========
class FLServer:
def __init__(self, consensus: ConsensusEngine,
blockchain: Blockchain, rbac: RBACContract):
self.consensus = consensus
self.blockchain = blockchain
self.rbac = rbac
self.global_model = IDSModel()

def gradient_verification(self, update: np.ndarray,
global_weights: np.ndarray) -> bool:
diff = np.linalg.norm(update - global_weights)
return diff < 5.0  # Simple poisoning threshold

def aggregate(self, updates: Dict[str, np.ndarray]):
total_cvw = sum(self.rbac.get_cvw(pk) for pk
in updates)
new_weights = np.zeros_like(self.global_model.
get_weights())

for pk, update in updates.items():
weight = self.rbac.get_cvw(pk) / total_cvw
new_weights += weight * update

self.global_model.set_weights(new_weights)

# ======== 7. Main Pipeline ========
def main():
# Initialize core components
blockchain = Blockchain()
rbac = RBACContract()
consensus = ConsensusEngine(consensus_type="PoS")
fl_server = FLServer(consensus, blockchain, rbac)

# Create IoT devices
devices = [
IoTDevice("D1", "Owner", cpu=3, memory=3,
energy=4),
IoTDevice("D2", "LegalUser", cpu=2, memory=2,
energy=3),
IoTDevice("D3", "Guest", cpu=1, memory=1,
energy=2)
]

# Registration phase
for d in devices:
rbac.register_device(d)
blockchain.log_event("DeviceRegistered",
d.device_id)
# FL Training Rounds
ROUNDS = 5
for r in range(ROUNDS):
print(f"\n--- Round {r+1} ---")
```

```
updates = {}

for d in devices:
if rbac.verify_device(d.public_key):
local_update = d.train_local(
fl_server.global_model.get_weights())

if fl_server.gradient_verification(local_update,
fl_server.global_model.get_weights()):
updates[d.public_key] = local_update
else:
blockchain.log_event("MaliciousUpdateRejected",
d.device_id)

# Consensus validation
if consensus.validate(updates):
fl_server.aggregate(updates)
blockchain.log_event("ModelAggregated",
f"Round {r+1}")
print(f"Round {r+1} aggregated successfully.")
else:
print(f"Round {r+1} failed consensus.")

# Final outputs
print("\n" + "="*50)
print("Final Global Model Weights:")
print(fl_server.global_model.get_weights())
print("\nBlockchain Ledger:")
blockchain.show_ledger()

# ======== 8. Entry Point ========
if __name__ == "__main__":
main()
```

A.2 Code Description

The above Python code implements a simplified version of the proposed blockchain-enhanced federated learning framework for smart home intrusion detection systems. The key components include:

- **IDSModel**: Lightweight neural network model for intrusion detection.
- **IoTDevice**: Represents smart home devices with cryptographic identities and computational volume weights (CVW).
- **RBACContract**: Smart contract implementation for role-based access control.
- **Blockchain**: Immutable ledger for recording all FL transactions and security events.
- **ConsensusEngine**: Implements PoS, PBFT, and PoA consensus mechanisms (selectable).
- **FLServer**: Central orchestrator for federated aggregation with gradient verification.

Note: This implementation is a proof-of-concept prototype. For production deployment, additional security measures, performance optimizations, and real-world dataset integration would be required.

Amro Alghamdi earned his Master of Science degree in Cybersecurity from AlMaarefa University in Riyadh, Saudi Arabia. His graduate research investigated the integration of blockchain technology to address critical security vulnerabilities in smart home surveillance networks. This work specifically targeted solutions for ensuring data integrity, preventing unauthorized access, and establishing tamper-proof audit logs within distributed IoT surveillance environments. (Email: 251130938@student.um.edu.sa)

Ismail Keshta received his B.Sc. and M.Sc. degrees in Computer Engineering, and his Ph.D. in Computer Science and Engineering, from King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2009, 2011, and 2016, respectively. From 2012 to 2016, he served as a lecturer in the Computer Engineering Department at KFUPM. Previously, in 2011, he was appointed as a lecturer at both Princess Nourah Bint Abdulrahman University and Imam Muhammad Ibn Saud Islamic University in Riyadh, Saudi Arabia. He is currently an associate professor in the Department of Computer Science and Information Systems at AlMaarefa University, Riyadh, Saudi Arabia, with active research interests in software process improvement, modeling, cryptography, cybersecurity, information security, and intelligent systems, reflecting his ongoing engagement in these areas. (Email: imohamed@um.edu.sa)