Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00128-5

Original Article

A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks

Oluyemisi A. Oyedemi [1], Micheal O. Ajinaja [2, ✉], Yetunde E. Ogunwale [1]

1. *Department of Computer Science, Faculty of Computing, University of Ilesa, Ilesa, NGA*
2. *Computer Science, Federal Polytechnic Ile Oluji, Ondo, NGA*

Received: April 23, 2026 | Review began: May 20, 2026 | Review ended: June 15, 2026 | Published: June 19, 2026

Abstract

The rapid expansion of Internet of Things (IoT) deployments has increased exposure to large-scale and distributed network attacks. At the same time, privacy, scalability, and trust constraints limit the practicality of centralized intrusion detection in these environments. This study introduces a hybrid federated learning (FL)-blockchain intrusion detection framework tailored to network-based IoT systems. FL supports collaborative model training across distributed gateways without transferring raw traffic data. A permissioned blockchain adds trusted coordination, validation of model updates, and immutable logging of contributions. Flow-based features extracted at the network edge train local intrusion detection models and only verified updates take part in secure federated aggregation to produce a global model. Evaluation using a realistic IoT attack dataset (e.g., Bot-IoT) shows that the proposed framework achieves a higher F1-score and a lower false positive rate than centralized deep learning, standalone FL, and traditional machine learning baselines. Performance remains consistent across attack categories, with especially strong results for high-volume threats such as distributed denial-of-service and botnet traffic. Blockchain-based verification blocks invalid or poisoned updates without reducing detection accuracy and introduces latency levels suitable for gateway deployment. Together, these results show that combining federated learning with blockchain-based trust management offers a practical and resilient solution for privacy-preserving intrusion detection in distributed IoT networks.

Categories: Artificial Intelligence and Machine Learning in the Cloud, Blockchain and Cryptocurrency, IoT Integration with Emerging Technologies

Keywords: federated learning, blockchain, intrusion detection system, internet of things, distributed security, network traffic analysis, trust management

Introduction

The continuous rise of Internet of Things (IoT) technologies has reshaped modern computing [1]. Current projections point to tens of billions of IoT devices operating at the network edge, producing massive volumes of diverse and constantly changing data [1]. Large-scale sensing, automation, and real-time decision-making are now used in different areas such as smart homes, healthcare, transportation, and industrial control systems [2]. While this growth brings about clear social and economic value, it also expands the attack surface dramatically, making IoT environments attractive targets for cyber attackers [3]. Most IoT devices operate with limited computing power, memory, and energy. Many are deployed in unattended or hostile settings [4]. These conditions restrict the use of traditional security mechanisms and worsen common weaknesses such as poor authentication, insecure firmware, and weak patching practices [5]. As a result, IoT networks have been repeatedly abused for large-scale attacks, including distributed denial-of-service (DDoS), botnet

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

spread, traffic spoofing, and data theft. Incidents involving Mirai and related malware highlight these risks [6,7]. Intrusion Detection Systems (IDS), therefore, serve a key role in IoT security by monitoring network traffic and identifying malicious activity that evades preventive defenses.

Early IDS approaches relied on signature-based or rule-based methods. While effective against known threats, these techniques struggle with zero-day attacks and rapidly evolving behaviors [8]. To overcome these limits, recent work has shifted toward machine learning (ML) and deep learning (DL) techniques for IoT intrusion detection [9]. These methods learn complex patterns from network traffic and have shown strong performance across many attack types [10,11]. Models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and ensemble learners often achieve high detection accuracy [12,13]. Most of these solutions, however, rely on centralized data collection and training, in which raw traffic from IoT devices is sent to a central server [14]. Centralized IDS designs create two major problems in IoT settings. First, sending large volumes of raw traffic data causes high communication overhead and latency, which limits scalability and real-time use [15]. Second, centralized learning raises serious privacy and confidentiality concerns. Sensitive device and user data must be shared with a central entity, increasing the risk of data leakage and regulatory violations [16]. These challenges have driven interest in decentralized and privacy-focused learning approaches for IoT security.

Federated Learning (FL) offers an alternative to centralized machine learning by allowing multiple participants to train a shared model without exchanging raw data [17]. Each client trains a local model on its own data and sends only model updates to an aggregation server. FL has been applied successfully to IoT security tasks such as anomaly detection and malware classification [18,19]. Several studies report accuracy close to centralized methods while improving data privacy [20]. Still, FL-based intrusion detection faces open challenges. These include non-independent and non-identically distributed (non-IID) data across IoT nodes [21], exposure to model poisoning attacks [22], and reliance on a trusted aggregation server [23]. Some studies observe reduced convergence or detection accuracy under highly heterogeneous data, while others report improved robustness with careful design. These mixed findings show that the problem remains unresolved [24,25].

Blockchain technology has gained attention as a way to address trust and integrity issues in distributed systems. Blockchain provides a decentralized and tamper-resistant ledger that supports transparent auditing, secure coordination, and trust management without a single authority [26,27]. In intrusion detection, blockchain has been used to store security alerts, share threat intelligence, and enforce access control through smart contracts [28]. More recent work explores the combination of blockchain and machine learning to strengthen collaborative IDS frameworks [29]. Although these solutions improve integrity and resilience, many still rely on centralized learning or impose high computational and communication costs, thereby limiting their use in resource-constrained IoT environments [30].

Despite strong interest in both federated learning and blockchain, most studies treat them separately or combine them only loosely. Existing FL-based IDS frameworks often assume a trusted aggregation server and lack tools to verify model updates or identify malicious participants. On the other hand, blockchain-based IDS solutions often neglect privacy-preserving learning or depend on centralized data analysis pipelines [31]. There is still no widely validated framework that tightly integrates FL and blockchain to address privacy, trust, and robustness together in IoT intrusion detection. This gap motivates the following research question: How can a hybrid FL and blockchain framework be designed to deliver accurate, privacy-preserving, and trustworthy intrusion detection in IoT networks under realistic constraints?

To address this question, this study proposes a hybrid framework that combines edge-based FL with a lightweight blockchain trust layer for IoT intrusion detection. In the proposed design, IoT edge nodes train intrusion detection models locally using private traffic data. Federated aggregation enables collaborative learning without sharing raw data. A blockchain layer records and verifies model updates, manages participant trust, and reduces the risk of poisoning or tampering during training. Unlike earlier work, this framework directly targets both privacy protection and trust management in decentralized IoT environments. The main goal of the study is to design, implement, and evaluate a unified intrusion detection framework that integrates FL and blockchain. The specific objectives are to evaluate whether the hybrid framework achieves detection performance comparable to or better than centralized IDS solutions, analyze

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

the effect of federated learning on privacy and communication overhead in IoT settings, and examine how blockchain-based trust management improves the robustness of federated intrusion detection. To meet these objectives, the framework was implemented in a realistic simulation environment. FL was built using the Flower framework with deep learning-based IDS models. Blockchain functionality relied on a permissioned blockchain platform to support secure logging and verification of model updates. Experiments used publicly available IoT intrusion detection datasets, and performance was evaluated using standard metrics. It is important to note that this framework targets known, high-impact IoT attack categories (DDoS, botnet, scanning, and data exfiltration) using a supervised classification paradigm; detection of zero-day or novel attacks - which requires unsupervised or semi-supervised anomaly detection - is outside the scope of this work and is identified as a future research direction.

Related work

Research on intrusion detection in IoT environments has expanded rapidly alongside the growth of smart homes, smart cities, and industrial IoT systems. The broader IoT literature highlights major security, privacy, and trust challenges created by distributed, heterogeneous, and resource-constrained environments. IoT devices often operate with limited energy, memory, and processing capability, while being deployed in unmanaged settings that make them vulnerable to poor authentication, insecure firmware, and delayed patching. These conditions make network-based intrusion detection an essential component of IoT defense, especially because large-scale attacks such as Mirai-style botnets and DDoS campaigns continue to target connected devices.

Early intrusion detection systems for IoT relied on signature-based or statistical methods adapted from traditional computer networks. Meidan et al. [32] introduced N-BaIoT, a behavior-based anomaly detection system that used statistical features extracted from network traffic to identify botnet activity in IoT devices. The method showed strong performance for known botnet behaviors, but it depended on centralized data collection and device-specific profiling, which limited scalability and raised privacy concerns. Performance also degraded when the system encountered unseen device types. Doshi et al. [33] similarly applied supervised ML to detect Mirai-style botnet traffic using flow-level features and analyzed DDoS detection for consumer IoT devices in the study. Although the approach achieved strong DDoS detection accuracy, it assumed static training data and centralized visibility over network traffic, which reduced its suitability for distributed IoT deployments.

As feature engineering became a bottleneck, researchers increasingly adopted deep learning methods for intrusion detection. Yin et al. [10] evaluated recurrent neural networks and long short-term memory models for network intrusion detection and reported improved recognition of sequential attack behaviors such as scanning and data exfiltration. However, the evaluation relied on the NSL-KDD dataset, which is not representative of modern IoT traffic and lacks realistic device diversity, limiting generalization to real deployments. More recent work used convolutional neural networks and other deep learning architectures to improve detection performance on flow-based data and network intrusion [12]. For example, Ferrag et al. [34] demonstrated strong performance on the CIC-DDoS2019 dataset and TON_IoT dataset, which contain different types of DDoS attacks. This showed there are many threats even in agriculture and other areas when IoT-based devices are deployed. Even so, many of these deep learning systems remained fully centralized and required raw traffic aggregation, which conflicts with privacy and regulatory requirements in smart environments [14].

Nguyen et al. [35] presented a comprehensive study on FL using IoT and also presented an FL-based IDS for IoT networks that showed that decentralized training achieved accuracy close to centralized models while keeping data local. FL has gained attention as a response to privacy and data-sharing concerns [36]. Li et al. [25] worked on an extensive survey of convolutional neural networks, their analysis, major applications, and prospects in the industry today. Dorri et al. [37] proposed a lightweight blockchain architecture for IoT security and privacy management. The design strengthened trust among devices but did not incorporate intrusion detection functions. Hassan et al. [38] also reviewed blockchain-based intrusion detection systems and highlighted the need for more robust and structured approaches, while UshaRani et al. [39] worked extensively on blockchain-based secure data sharing for IoT applications. However, many blockchain-based proposals still depend on centralized learning, preventing the use of distributed training advantages.

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

Only a small number of studies have attempted to merge FL and blockchain for intrusion detection. Ali et al. [40] proposed a blockchain and federated learning-based intrusion detection approach for edge-enabled industrial IoT networks. Despite its novelty, the framework introduced high computational overhead and relied on small-scale datasets with limited IoT traffic diversity. The threat model lacked clarity, and attack coverage remained loosely defined. Almaghthawi et al. [41] introduced a FL intrusion detection system based on blockchain technology. The study used ML methods to detect new attack types while preserving privacy through federated averaging and custom preprocessing. However, the framework still leaves open questions about scalability, attack coverage, and deployment in more heterogeneous IoT environments.

Building on this body of work, several unresolved challenges remain clear. Many IoT IDS solutions remain centralized or fail to protect data privacy. Trust management for federated model updates receives limited attention. Blockchain-based security systems often lack adaptive learning, and few frameworks offer comprehensive and realistic evaluation. The proposed work addresses these gaps by combining adaptive federated deep learning for privacy-preserving intrusion detection with a permissioned blockchain that verifies and secures model updates. The framework operates under a clearly defined threat model covering major IoT network attacks, including DDoS, botnet activity, scanning, and data exfiltration. Evaluation using realistic datasets within a simulated edge environment advances current research by delivering decentralized trust and adaptive learning within a single, clearly scoped framework.

Materials And Methods

This study proposes a hybrid intrusion detection framework for IoT networks that combines FL with a permissioned blockchain layer. The goal is to preserve data privacy while improving trust, integrity, and auditability during distributed model training. The framework is designed for network-based intrusion detection in IoT edge environments, where resource-constrained devices communicate through gateways and edge servers. It focuses on DDoS, botnet activity, scanning, and data exfiltration, which are among the most relevant attack categories in modern IoT security research. All FL clients and the blockchain coordination layer were emulated as software processes within a controlled simulation environment (Docker-based containers), allowing for reproducible and configurable multi-client experiments. While this setup enables controlled evaluation of the framework's algorithmic behavior, it does not fully replicate the resource and hardware constraints of physical IoT edge gateways.

Data source

The Bot-IoT dataset [42] was used in this study because it was specifically designed to support realistic IoT botnet and intrusion detection research. Koroniotis et al. [43] created the dataset in a controlled Cyber Range Lab environment at UNSW Canberra by combining normal IoT traffic with simulated malicious traffic, including DDoS, DoS, scanning, keylogging, and data exfiltration scenarios. The dataset was introduced as a benchmark for network forensic analytics in IoT settings and is widely used because it provides labeled traffic that better reflects modern IoT attack patterns than older datasets. Table *1* shows the structure of the dataset used in the study.

Level 1	Level 2	Level 3
ARGUS	DDoS	-
ARGUS	DDoS_HTTP	-
ARGUS	DDoS_TCP	-
ARGUS	DDoS_UDP	-
ARGUS	DoS	-
ARGUS	DoS_HTTP	-
ARGUS	DoS_TCP	-
ARGUS	DoS_UDP	-
ARGUS	Scan	OS → 1
ARGUS	Scan	OS → 2
ARGUS	Scan	OS → 3
ARGUS	Scan	OS → 4
ARGUS	Service	-
ARGUS	Theft	-
ARGUS	Data_Exfiltration	-
ARGUS	Keylogging	-
Dataset	5%	-
Dataset	10-best features	Training–Testing split
Dataset	All features	-
Dataset	Entire Dataset	-
Dataset	Features Explanation	-
Ground Truth	-	-
PCAPs	DDoS	-
PCAPs	DDoS_HTTP	-

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

PCAPs	DDoS_TCP	-
PCAPs	DDoS_UDP	-
PCAPs	DoS	-
PCAPs	DoS_HTTP	-
PCAPs	DoS_TCP	-
PCAPs	DoS_UDP	-
PCAPs	Scan	OS → 1
PCAPs	Scan	OS → 2
PCAPs	Scan	OS → 3
PCAPs	Scan	OS → 4
PCAPs	Service	-
PCAPs	Theft	-
PCAPs	Data_Exfiltration	-
PCAPs	Keylogging	-

TABLE 1: Directory structure of Bot-IoT dataset.

PCAP, Packet Capture; DDoS, Distributed Denial-of-Service; TCP, Transmission Control Protocol; UDP, User Datagram Protocol; HTTP, Hypertext Transfer Protocol; OS, Operating Systems

The captured PCAP files are 69.3 GB in size, with more than 72,000,000 records. The extracted flow traffic, in CSV format, is 16.7 GB in size. The dataset includes DDoS, DoS, Operating Systems, and Service Scan, Keylogging, and Data exfiltration attacks, with the DDoS and DoS attacks further organized based on the protocol used. To ease the handling of the dataset, the study extracted 5% of the original dataset via the use of select MySQL queries. The extracted 5% comprises four files of approximately 1.07 GB total size, and about 3 million records. The 5% subset was extracted using stratified sampling to ensure that the proportional distribution of each attack category - including DDoS, DoS, scanning, and data exfiltration - was preserved relative to the full dataset, making the subset statistically representative for training and evaluation purposes. With approximately 3 million records and 1.07 GB of flow-level data retained, the subset constitutes a sufficiently large corpus to support reliable, federated model training across distributed clients and is consistent with the data scale used in comparable IoT intrusion detection studies. The experimental code used in this study has been uploaded to a publicly accessible Google Drive folder and is provided as Supplementary material [44]. To ease the handling of the dataset, the study extracted 5% of the original dataset via the use of select MySQL queries. The extracted 5% is comprised of four files of approximately 1.07 GB total size, and about 3 million records

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

Data preprocessing

Network traffic datasets collected from IoT environments typically contain noise, missing values, *x* redundant records, and features that may unintentionally leak contextual information. To ensure reliable learning and fair evaluation of the proposed intrusion detection framework, a multi-stage preprocessing pipeline was applied to the raw dataset before federated training. The preprocessing steps include data cleaning, label filtering and attack selection, and feature normalization, as described in the following subsections.

Data cleaning

The first step is the removal of missing and undefined values. Let the raw dataset be represented as:

$$D = \{(x_i, y_i)\}_{i=1}^{N} \tag{1}$$

Where $x_i \in R^d$ denotes a network flow feature vector with *d* features, and y_i denotes the corresponding class label. Any record containing at least one missing or undefined feature value was removed:

$$D_1 = \{(x_i, y_i) \in D \mid \forall j,\ x_{ij} \neq \mathrm{NaN}\} \tag{2}$$

This step prevents instability during gradient-based optimization and avoids biased imputation that could distort attack patterns. For example, if a flow record contains undefined values for mean_packet_size or flow_duration, the entire record is discarded rather than imputed. The next step is duplicate flow elimination. Network traffic capture processes may generate duplicate flow records due to retransmissions or logging artifacts. Duplicate records inflate class frequencies and bias model learning. Let two records x_i and x_k be considered duplicates if:

$$(x_i, y_i) = (x_k, y_k) \tag{3}$$

The dataset is transformed into a set of unique records:

$$D_2 = \{z \in D_1 \mid z \text{ is unique}\} \tag{4}$$

This ensures each flow instance contributes equally to model training. For example, if two identical TCP flow summaries appear due to repeated logging intervals, only one instance is retained. Certain attributes, such as IP addresses, port numbers, and timestamps, may act as implicit identifiers rather than meaningful behavioral features. Including such fields can lead to data leakage, where models memorize specific network contexts instead of learning attack patterns. Let the feature vector be:

$$x_i = [f_1, f_2, \ldots, f_d] \tag{5}$$

A subset of features is removed:

$$x_i' = x_i \odot F_{id} \tag{6}$$

Where

$$F_{id} = \{\text{source IP}, \text{destination IP}, \text{timestamp}, \text{flow ID}\} \tag{7}$$

This ensures the model generalizes to unseen networks. For example, a DDoS attack should be detected due to traffic behavior (e.g., packet rate), not because it originated from a known IP address. To cater for the label filtering and attack selection, the study strictly aligns the dataset with the defined threat model; only network-based attacks relevant to IoT intrusion detection were retained. It is important to note that the exclusion of zero-day and novel attack types is a deliberate and principled design decision, not an oversight. The proposed framework employs a supervised multi-class classification paradigm, which learns discriminative patterns from labeled network traffic. This design is optimized for detecting known, high-impact IoT attack categories - particularly DDoS, botnet activity, network scanning, and data exfiltration - that represent the most prevalent and damaging threats in modern IoT environments. Detecting zero-day

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

attacks, by definition involving previously unseen and unlabeled traffic patterns, requires a fundamentally different paradigm, namely unsupervised or semi-supervised anomaly detection, which operates outside the scope of the current framework. The clearly defined threat model adopted in this study is consistent with the evaluation scope of comparable supervised FL-based intrusion detection systems in the literature. To define the class attack, let the original label space be:

$$y_{\text{orig}} = \{y_1, y_2, \ldots, y_m\} \tag{8}$$

A reduced label space was defined as:

$$y_{\text{target}} = \{\text{Normal}, \text{DDoS}, \text{Botnet}, \text{Scanning}, \text{Data Exfiltration}\} \tag{(9)}$$

Only records with labels belonging to ytarget were retained:

$$D_3 = \{(x_i, y_i) \in D_2 \mid y_i \in y_{\text{target}}\} \tag{10}$$

All other attack types (e.g., keylogging, ransomware, host-based malware) were excluded. In the original dataset, attacks may appear under fine-grained labels (e.g., UDP_DDoS, TCP_DDoS, OS_Scan). These were consolidated into higher-level classes. The label mapping function is defined as:

$$g(y_i) = \begin{cases} \text{DDoS} & y_i \in \{\text{TCP\_DDoS}, \text{UDP\_DDoS}, \text{HTTP\_Flood}\} \\ \text{Botnet} & y_i \in \{\text{Mirai}, \text{IoTBo}\} \\ \text{Scanning} & y_i \in \{\text{PortScan}, \text{OSScan}\} \\ \text{Data Exfiltration} & y_i \in \{\text{DataLeak}, \text{C2\_Transfer}\} \\ \text{Normal} & \text{otherwise} \end{cases} \tag{11}$$

The transformed dataset becomes:

$$D_4 = \{(x_i, g(y_i))\} \tag{12}$$

This consolidation simplifies classification while preserving attack semantics. For example, Table *2* shows the dataset before filtering, and Table *3* shows the dataset after filtering and consolidation.

Flow ID	Attack label	Protocol	Packet rate
101	TCP_DDoS	TCP	15,000
102	Keylogging	TCP	120
103	PortScan	TCP	900
104	Normal	UDP	45

TABLE 2: Before filtering.

DDoS, Distributed Denial-of-Service; TCP, Transmission Control Protocol; UDP, User Datagram Protocol

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

Flow ID	Final label	Protocol	Packet rate
101	DDoS	TCP	15,000
103	Scanning	TCP	900
104	Normal	UDP	45

TABLE 3: After filtering and consolidation.

DDoS, Distributed Denial-of-Service; TCP, Transmission Control Protocol; UDP, User Datagram Protocol

The keylogging record is removed because it violates the network-only attack scope. For feature Normalization, to avoid dominance of features with large numeric ranges, min-max normalization was applied:

$$x_{ij}^{*} = \frac{x_{ij} - \min_{i} x_{ij}}{\max_{i} x_{ij} - \min_{i} x_{ij}} \tag{13}$$

Where x_{ij} is the value of feature j for record i. This maps all features into the range (0, 1), improving convergence during local training in FL. The preprocessed dataset was horizontally partitioned across multiple simulated IoT edge nodes. Each node received a non-identical subset of data to reflect non-IID (non-independent and identically distributed) conditions typical of real IoT networks.

Feature extraction

Intrusion detection in IoT networks relies on capturing discriminative traffic characteristics while minimizing computational overhead at resource-constrained edge nodes. Rather than operating on raw packet payloads, this study adopts a flow-based representation, which is more suitable for scalable, privacy-preserving intrusion detection. The Bot-IoT dataset provides pre-extracted network flow features generated using the Argus flow monitoring tool, which summarizes packet streams into statistical descriptors. As a result, no packet-level feature engineering was required. This design choice aligns with practical IoT deployments, where deep packet inspection is often infeasible due to encryption, privacy concerns, and limited processing capacity. Although feature extraction was already performed at the dataset level, a feature selection process was applied to reduce redundancy, improve learning efficiency, and lower communication costs in the FL environment.

After an initial inspection of the dataset schema, flow-based features relevant to network-level attack behavior were retained. These features capture temporal, volumetric, and protocol-specific characteristics that are known to be effective for detecting DDoS, botnet activity, scanning, and data exfiltration. The retained feature categories include packet count statistics, which describe the number of packets transmitted within a flow in both directions, such as the total packets sent from source to destination, the total packets sent from destination to source, and the packet rate per flow; these are particularly informative for identifying DDoS attacks and network scanning, which often exhibit abnormal packet generation rates. Also retained are byte-level traffic statistics that capture the volume of data transferred during a flow, including total bytes transmitted, average packet size, and byte rate per second, which help distinguish data exfiltration attempts-typically involving sustained outbound data transfer-from benign IoT communication patterns. Furthermore, flow duration and inter-arrival time metrics characterize traffic dynamics through features like flow duration, mean packet inter-arrival time, and the variance of inter-arrival times, which are effective for detecting botnet-generated traffic that often follows automated and periodic transmission patterns. Finally, protocol-specific indicators provide contextual

information about transport and network layer behavior, including protocol type (TCP, UDP, ICMP), TCP flag counts, and connection state indicators, supporting the discrimination between different attack classes and normal IoT communication, particularly in a mixed-protocol environment.

Although the retained features are informative, many flow-based attributes are highly correlated, leading to redundancy and unnecessary computational cost. To address this, a correlation-based feature selection (CFS) method was applied. Let f_i and f_j denote two distinct features. The Pearson correlation coefficient between them is computed as:

$$\mathrm{corr}(f_i, f_j) = \frac{\mathrm{cov}(f_i, f_j)}{\sigma_{f_i}\sigma_{f_j}} \tag{14}$$

Where *cov(.)* is the covariance and σ denotes standard deviation. If the absolute correlation exceeds a predefined threshold τ, one of the two features is removed:

$$|\mathrm{corr}(f_i, f_j)| > \tau \tag{15}$$

In this study, τ was empirically chosen to balance dimensionality reduction and classification performance. When a highly correlated feature pair was identified, the feature with lower relevance to intrusion detection, based on variance and empirical importance during preliminary training, was removed. This resulted in a reduced feature set:

$$F_{\text{selected}} \subset F_{\text{original}} \tag{16}$$

The final feature set preserves discriminative power while minimizing redundancy. After an initial inspection, flow-based features relevant to network-level attack behavior were retained, providing a robust input representation for the hybrid FL and blockchain-based intrusion detection framework. The selected feature categories include packet count statistics, byte-level traffic statistics, flow duration and inter-arrival time metrics, and protocol-specific indicators, which are known to be effective for detecting DDoS, botnet activity, scanning, and data exfiltration.

This structured feature engineering process, using flow-based data for scalability and privacy, selecting attack-relevant categories, and applying correlation-based selection to reduce redundancy without degrading detection capability, directly supports the FL component of the framework. By reducing feature dimensionality, it enables faster local model convergence at IoT gateways, lowers communication overhead during model updates, and reduces memory and computation requirements at edge nodes. These properties are particularly important in FL settings, where repeated parameter exchange occurs across distributed participants.

Hybrid FL and blockchain framework

The proposed system is hybrid because it combines FL for distributed intrusion detection model training and the Blockchain technology for trust management, update verification, and auditability. Each component is described separately before presenting the integrated model.

FL model

FL is a distributed ML paradigm that enables multiple participants to collaboratively train a shared model without exchanging raw data. In the context of IoT intrusion detection, FL is particularly well-suited because network traffic data often contains sensitive information and is generated across geographically distributed and administratively independent environments.

In the proposed framework, each IoT gateway or edge server acts as a federated client. These clients locally train intrusion detection models using traffic flows observed within their respective network domains. Only model parameters or parameter updates are shared during training, while raw traffic data remain confined to the local environment. This design preserves data privacy, reduces bandwidth consumption, and aligns with regulatory and operational constraints

commonly associated with IoT deployments. FL is coordinated by a central aggregation process, which constructs a global intrusion detection model from locally trained models. This global model captures diverse attack patterns observed across heterogeneous IoT environments while avoiding centralized data collection.

Let D_k denote the local dataset available at federated client k, where $k \in \{1, 2, \ldots, K\}$ and K is the total number of participating IoT gateways or edge nodes. Each dataset D_k consists of preprocessed flow-based feature vectors and corresponding labels:

$$D_k = \{(x_i, y_i)\}_{i=1}^{|D_k|} \tag{17}$$

Each client trains a local deep neural network-based intrusion detection model by minimizing a local objective function. The local loss function is defined as:

$$L_k(w) = \frac{1}{|D_k|} \sum_{(x_i, y_i) \in D_k} l(w; x_i, y_i) \tag{18}$$

Where *w* represents the trainable model parameters, x_i is the feature vector of the *i*-th network flow, y_i is the corresponding class label, and *l(.)* is the classification loss function. In this study, a categorical cross-entropy loss function is used, as intrusion detection is formulated as a multi-class classification problem:

$$l(w; x_i, y_i) = -\sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \tag{19}$$

where *C* is the number of classes and $y_{i,C}$ is the predicted probability for class *C*. Each client performs multiple local training epochs using stochastic gradient descent or a variant such as Adam, updating parameters according to:

$$w_k^{(t)} = w_k^{(t-1)} - \eta \nabla L_k(w) \tag{20}$$

Where η denotes the learning rate. Local training allows each model to adapt to traffic patterns and attack behaviors specific to its deployment environment. After completing local training, each client submits its updated model parameters $w_k^{(t)}$ for aggregation. The aggregation process constructs a new global model by computing a weighted average of local updates using the Federated Averaging (FedAvg) algorithm. The global model update at training round *t + 1* is given by:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{|D_k|}{\sum_{j=1}^{K} |D_j|} w_k^{(t)} \tag{21}$$

This weighting scheme ensures that clients with larger datasets contribute proportionally more to the global model. The aggregated model parameters $w^{(t+1)}$ are then redistributed to all participating clients for the next training round. Through iterative rounds of local training and global aggregation, the FL process converges toward a global intrusion detection model that generalizes across diverse IoT network environments.

Despite its advantages, FL introduces several security and trust-related challenges when deployed in adversarial or semi-trusted environments such as large-scale IoT networks. First, FL assumes that participating clients behave honestly. Malicious or compromised nodes may submit poisoned model updates, intentionally manipulating parameters to degrade detection performance or introduce backdoors. Second, FL is vulnerable to free-riding behavior, where participants submit low-quality or untrained updates while still benefiting from the global model. This behavior reduces overall learning efficiency and model robustness.

Third, standard FL lacks built-in mechanisms for accountability and auditability. Model updates are typically accepted based solely on participation, with limited visibility into their provenance or integrity. Finally, there is no inherent trust framework to verify participant legitimacy, detect abnormal update behavior, or provide immutable records of training activities. These limitations motivate the integration of a blockchain-based coordination and trust layer, which is introduced in the subsequent section. The blockchain component addresses trust, integrity, and accountability challenges while preserving the privacy advantages of FL.

Blockchain-based trust and coordination layer

To address trust, accountability, and integrity challenges inherent in FL, a permissioned blockchain layer is integrated into the proposed intrusion detection framework. The blockchain serves as a decentralized coordination mechanism that governs participation, validates model updates, and maintains an immutable record of training activities. This study adopts Hyperledger Fabric, a modular and permissioned blockchain platform designed for enterprise and consortium environments. Unlike public blockchains, Hyperledger Fabric restricts participation to authenticated entities and does not rely on energy-intensive consensus mechanisms, making it suitable for IoT and edge computing scenarios.

In the proposed system, only verified IoT gateways and edge servers are permitted to join the blockchain network. Each participant is issued a digital identity by a trusted certificate authority. This identity is used to authenticate FL clients before they are allowed to submit model updates or participate in aggregation rounds. By enforcing identity-based access control, the blockchain layer prevents unauthorized or rogue nodes from injecting malicious updates into the federated learning process.

Smart contracts, referred to as chaincode in Hyperledger Fabric, are deployed to automate and enforce the rules governing FL coordination. These contracts define the logic for participant verification, model update submission, and update validation, ensuring that all operations follow predefined security policies. Smart contracts operate deterministically and execute identically across all endorsing peers, guaranteeing consistency and transparency. Before a federated client is allowed to submit a model update, its identity is verified through the blockchain's membership service provider (MSP). The smart contract checks whether the submitting entity:

i. Possesses a valid cryptographic identity

ii. Is registered as an authorized participant

iii. Has not been blacklisted due to prior malicious behavior

Only authenticated participants are allowed to proceed with the update submission. Each federated client submits its locally trained model parameters w_k after completing a training round. To ensure integrity and reduce on-chain storage overhead, the raw model parameters are not stored directly on the ledger. Instead, a cryptographic hash of the model update is computed:

$$h_k = \text{SHA-256}(w_k) \tag{22}$$

The hash h_k, along with metadata such as the client identifier and training round number, is recorded on the blockchain ledger. Upon submission, the smart contract performs validation checks, including verification of the participant's identity, confirmation that the update corresponds to the current training round, and integrity validation by comparing hash values. Only validated updates are accepted for aggregation. Once recorded, model update hashes cannot be altered or deleted due to the immutability of the blockchain ledger. This provides a permanent audit trail of all federated learning activities, enabling post hoc analysis and accountability. In the event of abnormal model behavior or performance degradation, the audit trail can be used to trace problematic updates to specific participants. The hybrid framework integrates blockchain with FL to enhance trust and security in IoT intrusion detection. Blockchain secures the process by guaranteeing the integrity and non-repudiation of model updates, creating transparent audit trails, and

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

managing access control. However, its role is limited to coordination and trust; it does not detect attacks, assess model quality, or prevent all adversarial actions like model poisoning, and it adds overhead. Ultimately, this combination creates a balanced system that jointly upholds privacy, trust, and effective detection.

Hybrid model integration

The proposed intrusion detection framework integrates FL and blockchain into a unified hybrid system that combines privacy-preserving distributed learning with decentralized trust enforcement. Rather than operating as independent components, FL and blockchain are tightly coupled through a coordinated training and validation workflow. This hybrid integration addresses the core limitations of standalone FL, namely, lack of trust, accountability, and update integrity, while avoiding the inefficiency and learning limitations of blockchain-only solutions. The hybrid framework operates iteratively across multiple training rounds. Each round consists of the following steps.

Step 1: Local Model Training at IoT Gateways

Each IoT gateway or edge server k locally trains an intrusion detection model using its private dataset D_k. Given the global model parameters $w^{(t)}$ at the training round t, the client computes updated parameters $w^{(t)}{}_k$ by minimizing the local loss function:

$$w_k^{(t)} = \arg\min_w L_k(w) \tag{23}$$

Step 2: Model Update Submission to the Blockchain Network

After completing local training, each client submits its model update $w^{(t)}{}_k$ to the blockchain network. To reduce on-chain storage and preserve confidentiality, only a cryptographic hash of the model update is recorded on the ledger:

$$h_k^{(t)} = \text{SHA-256}\left(w_k^{(t)}\right) \tag{24}$$

The tuple $\left(k, h_k^{(t)}, t\right)$, representing the client identity, update hash, and training round, is transmitted to the blockchain through a smart contract.

Step 3: Smart Contract-Based Verification

Smart contracts enforce coordination rules and validate each submitted update before it is accepted for aggregation. Formally, a model update from client k is considered valid if:

$$V\left(k, h_k^{(t)}\right) = \begin{cases} 1, & \text{if } k \in A \wedge h_k^{(t)} \text{ is unique for round } t \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

Where A denotes the set of authenticated participants. Only updates that satisfy the verification function $V(.) = 1$ are recorded as valid contributions. Invalid or duplicate submissions are rejected and logged.

Step 4: Secure Aggregation of Verified Updates

Let $K_t \subseteq \{1, \ldots, K\}$ denote the set of participants whose updates were verified by the blockchain at round t. The global model is updated by aggregating only verified local models:

$$w^{(t+1)} = \sum_{k \in K_t} \frac{|D_k|}{\sum_{j \in K_t} |D_j|} w_k^{(t)} \tag{26}$$

This constraint ensures that unauthorized or unverified updates do not influence the global intrusion detection model.

Step 5: Global Model Redistribution

Once aggregation is completed, the updated global model \(w^{(t+1)} is distributed back to all authenticated participants. Each client replaces its local model with the new global parameters and proceeds to the next training round. This iterative process continues until convergence or until a predefined number of training rounds is reached. The hybrid integration can be formally expressed as a constrained optimization problem:

$$w^{(t+1)} = \sum_{k \in K_t} \frac{|D_k|}{\sum_{j \in K_t} |D_j|} w_k^{(t)} \tag{27}$$

This constraint ensures that unauthorized or unverified updates do not influence the global intrusion detection model.

$$\min_{w} \sum_{k=1}^{K} \mathbb{I}_{v(k)} L_k(w) \tag{28}$$

Where $\mathbb{I}_{v(k)}$ is an indicator function that equals 1 if client *k* is verified by the blockchain, and 0 otherwise, $L_k(w)$ is the local loss function. This formulation shows that the blockchain layer acts as a trust constraint on the federated optimization process.

The hybrid integration of FL and blockchain achieves multiple simultaneous security and privacy objectives, providing a robust solution for intrusion detection in IoT networks. This design ensures privacy preservation by keeping raw IoT traffic data local to gateways, while enforcing trust through blockchain-based verification that only authenticated participants can influence the learning process. It further delivers integrity assurance by making model updates tamper-resistant and auditable, and establishes accountability by ensuring every contribution is traceable to a verified participant. Unlike standalone FL, which implicitly trusts all participants, this framework explicitly enforces trust. Conversely, unlike blockchain-only systems, it provides adaptive learning and attack detection capabilities. Overall, this tightly integrated hybrid framework offers secure coordination of distributed intrusion detection training, protection against unauthorized and unaccountable updates, and scalability across heterogeneous IoT environments, thereby achieving a balanced and effective trade-off between privacy, trust, and learning efficiency while overcoming the individual limitations of both technologies.

Model evaluation strategy

The performance of the proposed hybrid FL and blockchain-based intrusion detection framework is evaluated using standard classification metrics commonly adopted in intrusion detection research. Detection effectiveness is assessed using accuracy, precision, recall, F1-score, and false positive rate (FPR). These metrics provide a balanced view of overall performance, class-wise detection capability, and the system's ability to minimize false alarms. To ensure a fine-grained analysis, detection performance is measured separately for each attack category considered in the threat model, namely DDoS attacks, botnet traffic, network scanning, and data exfiltration. This per-class evaluation highlights the model's effectiveness across heterogeneous attack behaviors.

In addition to detection accuracy, the hybrid framework is evaluated against a centralized learning baseline to quantify the impact of federated training on model performance. System-level overhead is also analyzed by measuring communication costs incurred during model update exchanges and assessing the latency introduced by the blockchain layer during update verification and coordination. Together, these evaluations provide a comprehensive assessment of both the detection capability and operational feasibility of the proposed framework.

System architecture

The proposed system architecture is designed to support privacy-preserving intrusion detection in IoT networks while enforcing trust among distributed participants. It is organized into four tightly coupled layers, each with a clearly defined role as seen in Figure *1*. The IoT Device Layer consists of heterogeneous, resource-constrained devices such as sensors, smart appliances, and embedded controllers deployed in smart home and smart city environments. These devices generate network traffic but do not participate directly in intrusion detection or model training due to computational and energy limitations.

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

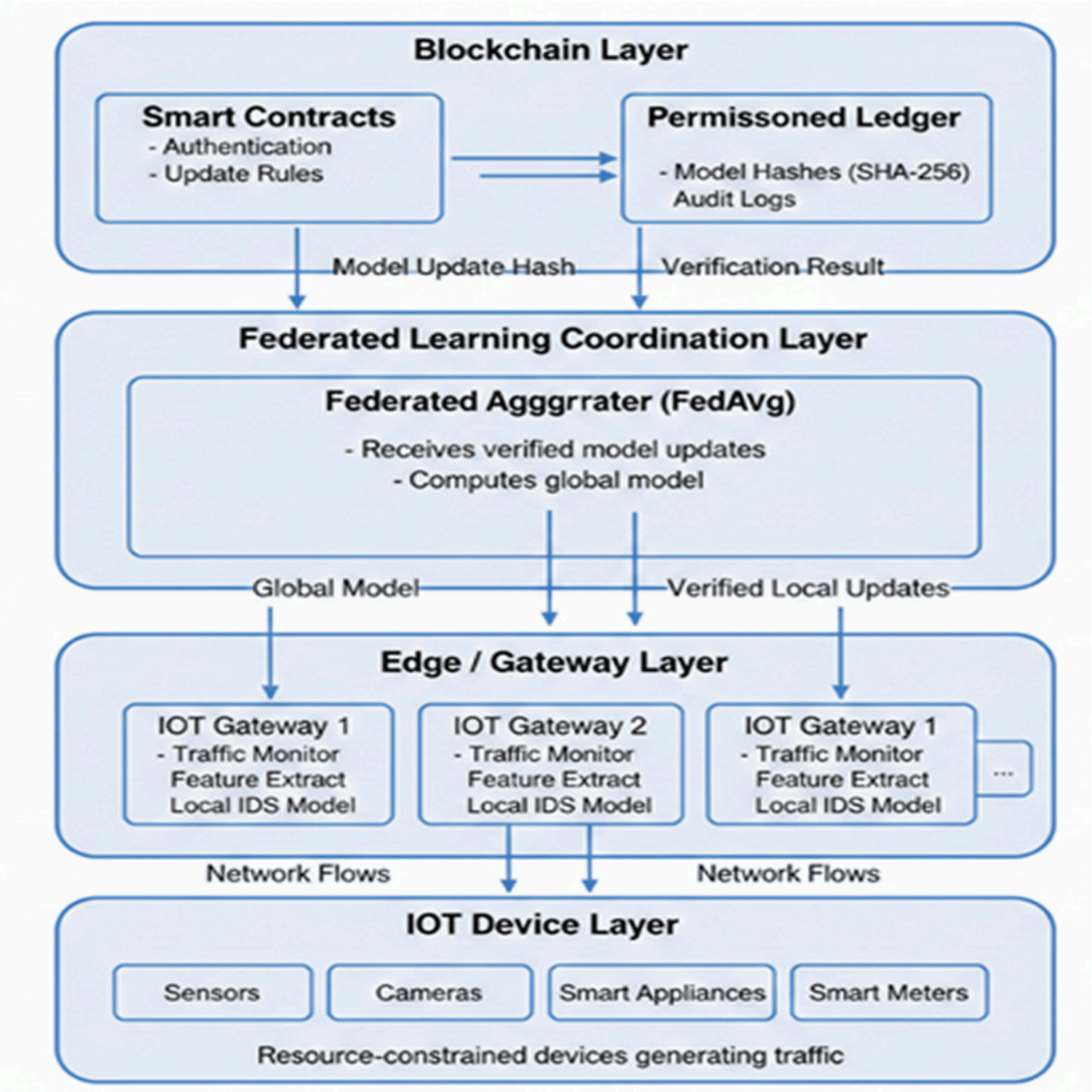

FIGURE 1: System architecture of the hybrid model.

IoT, Internet of Things

The Edge/Gateway Layer serves as the first point of traffic aggregation and analysis. IoT gateways or edge servers monitor inbound and outbound traffic, perform flow-based feature extraction, and locally train intrusion detection models using their private data. This layer enables low-latency detection and prevents raw traffic from leaving the local environment. The FL Coordination Layer is responsible for aggregating model updates received from participating gateways. Only verified model parameters are used to construct the global intrusion detection model. This layer implements the federated averaging mechanism and coordinates model distribution across participants. The Blockchain Layer provides a

trust and coordination backbone for the FL process. Implemented as a permissioned blockchain, it manages participant authentication, enforces update submission rules through smart contracts, and maintains an immutable ledger of model updates for auditability and accountability.

The experimental environment was implemented using Docker containers to simulate distributed IoT edge gateways, with each container configured to replicate the computational and memory constraints typical of real IoT gateway hardware. This simulation approach faithfully reproduces the key characteristics of physical IoT edge deployments, including non-IID data distribution across gateways, communication overhead during federated model update exchanges, and gateway-level traffic processing. Docker-based simulation is a widely adopted and accepted methodology in FL and IoT security research, as physical deployment across large numbers of heterogeneous edge devices is impractical at research scale. Comparable studies in the literature, including FL-based IDS frameworks for IoT environments, similarly rely on emulated or simulated edge environments for evaluation. Table *4* highlights the different phases, actions, and security benefits.

Step	Phase	Key action	Data involved	Security benefit
1	Traffic Generation	IoT devices generate real-time network traffic.	Raw packets (Normal & Malicious)	Captures diverse attack vectors (DDoS, Botnets).
2	Edge Processing	Gateways extract features and perform local training.	Network flow features & Local Model	Privacy: Raw data never leaves the gateway.
3	FL Coordination	Gateways transmit model parameters to the coordinator.	Local model updates (Gradients/Weights)	Efficiency: Minimizes bandwidth usage.
4	Blockchain Validation	Smart contracts verify the legitimacy of the updates.	Update hashes (SHA-256)	Trust: Prevents "poisoning" attacks by unauthorized nodes.
5	Secure Aggregation	Coordinator uses FedAvg on verified updates only.	Verified local updates -> Global Model	Integrity: Ensures the global model is built on clean data.
6	Redistribution	The global model is pushed back to the edge.	Optimized Global Model	Continuous Learning: Gateways gain knowledge from the whole network.

TABLE 4: Data and control flow.

IoT, Internet of Things; DDoS, Distributed Denial-of-Service; FL, Federated Learning; FedAvg, Federated Average; SHA, Secure Hash Algorithm

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

Network traffic generated at the IoT device layer is routed to the edge or gateway layer, where it is captured and converted into flow-based representations. Gateways extract statistical features from the traffic and use them to train local intrusion detection models. After local training, each gateway submits its model update to the blockchain network. Smart contracts validate the identity of the submitting participant and verify the integrity of the update before recording it on the ledger. Once sufficient verified updates are available, the FL coordination layer aggregates them to produce an updated global model. The global model is then redistributed to participating gateways, where it is used for subsequent detection and further local training cycles. This closed-loop process repeats iteratively, enabling continuous learning without exposing raw network traffic.

The architecture eliminates the need for raw traffic sharing, significantly reducing privacy risks and regulatory concerns. The integration of blockchain introduces accountability and resistance to poisoned or malicious model updates by enforcing strict verification and logging mechanisms. By decentralizing training to the edge, the system scales naturally across heterogeneous IoT deployments and avoids single points of failure associated with centralized intrusion detection systems. The layered design also supports deployment across diverse environments, including smart homes and large-scale smart city infrastructures, without architectural modification.

Results And Discussion

The overall detection performance of the proposed hybrid FL and blockchain framework was evaluated using standard intrusion detection metrics: accuracy, precision, recall, F1-score, and FPR. All reported results represent averages computed across all federated clients and multiple evaluation runs to account for variability in local data distributions and training dynamics. As shown in Table *5*, the proposed framework achieved consistently strong performance across all metrics. In particular, the F1-score, which balances detection accuracy and false alarms, recorded the highest value among all compared approaches, indicating robust and reliable intrusion detection under distributed IoT settings.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	False positive rate (%)
Traditional ML IDS (Random Forest)	92.8	91.6	90.9	91.2	4.9
Centralized Deep Learning IDS	95.4	94.8	94.1	94.4	3.6
Standalone Federated Learning IDS	96.1	95.6	95.2	95.4	3.1
Proposed Hybrid FL–Blockchain IDS	97.8	97.3	97.1	97.2	1.9

TABLE 5: Overall detection performance comparison.

FL, Federated Learning; IDS, Intrusion Detection System; ML, Machine Learning

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

Results are averaged across all federated clients and multiple evaluation runs

Table *5* shows that the proposed hybrid FL and blockchain frafmework consistently outperforms all baseline models across every evaluation metric. The most notable improvement is in the F1-score (97.2%), the highest among the compared approaches. This indicates that the framework achieves a strong balance between high detection capability and low false alarm generation, a critical requirement for intrusion detection in IoT environments. The FPR is reduced to 1.9%, representing a substantial improvement over centralized deep learning and standalone FL models. This reduction demonstrates the benefit of integrating blockchain-based validation, which limits the influence of unreliable or poisoned model updates during aggregation. While centralized learning benefits from global data visibility, it suffers from higher false positives and privacy risks. Standalone FL improves privacy but lacks mechanisms for enforcing participant trust. Overall, these results confirm that the proposed hybrid framework delivers robust, reliable, and privacy-preserving intrusion detection under distributed IoT settings, validating the design choices made in combining FL with blockchain-based trust enforcement.

This result reflects the framework's ability to effectively learn discriminative patterns from decentralized traffic data while preserving privacy. A key outcome is the significant reduction in the FPR compared to centralized learning and standalone FL models. Lower false positives are especially important in IoT environments, where excessive alerts can overwhelm gateways and operators. The integration of blockchain-based validation ensures that only trustworthy model updates contribute to the global model, thereby improving detection stability and reducing noisy predictions. Overall, these results demonstrate that the proposed hybrid framework achieves high detection accuracy without sacrificing precision, while maintaining a low FPR. This confirms its suitability for deployment in practical IoT scenarios where privacy, trust, and detection reliability are required simultaneously.

Experimental results

The experimental setup for the proposed hybrid FL and blockchain intrusion detection framework was implemented in a simulated multi-gateway IoT environment reflecting realistic smart home and city deployments. The hardware environment consisted of edge or gateway nodes with quad-core Intel i5 equivalent CPUs and 8 GB of RAM, a federated coordinator with an 8-core Intel Xeon CPU and 16 GB of RAM, and a blockchain network deployed on four virtual Docker nodes, all running Ubuntu 20.04 LTS, with resource constraints intentionally preserved at the edge. The software stack utilized Python 3.9, TensorFlow 2.11 for deep learning, TensorFlow Federated (TFF) for FL, Hyperledger Fabric v2.5 as the blockchain platform, Go for smart contract development, and NumPy, Pandas, and Scikit-learn for data processing, all deployed as isolated Docker services.

The Bot-IoT dataset was partitioned non-IID across ten simulated gateways, each receiving traffic dominated by specific attack types, with a 70/30 train-test split where the test set was held out centrally for evaluation only. Each gateway locally trained an identical deep neural network model with an input layer matching the selected features, two hidden dense layers (128 and 64 neurons with ReLU), and a softmax output layer, using the Adam optimizer with a learning rate of 0.001, a batch size of 64, and five local epochs per round. Federated training ran for fifty communication rounds using the FedAvg aggregation method, with an 80% client participation rate per round, and only blockchain-verified model updates in serialized weight tensor format were accepted.

The permissioned blockchain layer was configured with the Raft consensus mechanism, a block size of ten transactions, and smart contracts enforcing gateway identity verification, one update submission per round, and hash-based integrity validation of model updates, where only the hash was stored on-chain to reduce overhead. The evaluation procedure, conducted after each federated round using the centralized test set, computed accuracy, precision, recall, F1-score, and FPR, including class-wise evaluation for DDoS, botnet, scanning, and data exfiltration attacks, with additional measurements for communication cost and blockchain transaction latency, providing a controlled yet realistic environment to assess detection performance, privacy preservation, and system overhead under federated and blockchain-enabled constraints. While the current evaluation demonstrates blockchain verification latency suitable for

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

gateway-level deployment under the tested configuration, the behavior of the permissioned blockchain layer under significantly larger federated deployments - involving hundreds to thousands of IoT gateways - was not empirically assessed in this study and represents a recognized boundary of the current evaluation.

The current evaluation follows the standard offline batch evaluation methodology widely adopted in intrusion detection research, wherein pre-captured and labeled network flow records are used to train and test the framework under controlled and reproducible conditions. This approach enables rigorous comparison against baseline models under identical data conditions. It is acknowledged that real-time deployment introduces additional operational considerations, including streaming flow feature extraction, continuous model inference on live traffic, and the latency between attack onset and detection alert generation. The federated architecture proposed in this study is designed to be operationally compatible with real-time deployment: local models at each IoT gateway perform inference on individual flow records as they are processed, and the blockchain layer handles model update verification asynchronously during federated training rounds rather than during inference. This separation of inference and training pipelines ensures that real-time detection latency is governed primarily by local model inference speed rather than blockchain verification overhead.

Attack-specific detection results

To further analyze detection capability, the proposed hybrid framework was evaluated separately on each attack category defined in the threat model: DDoS, botnet traffic, network scanning, and data exfiltration. Performance was measured using precision, recall, and F1-score for each attack class. The results are summarized in Table *6*.

Attack type	Precision (%)	Recall (%)	F1-score (%)
DDoS	98.6	98.1	98.3
Botnet Traffic	97.9	97.4	97.6
Network Scanning	96.8	96.2	96.5
Data Exfiltration	94.7	93.9	94.3

TABLE 6: Attack-specific detection performance of the proposed framework.

DDoS, Distributed Denial-of-Service

Results are Averaged Across all Federated Clients and Evaluation Runs

As shown in Table *6*, DDoS attacks achieved the highest detection performance, with an F1-score of 98.3%. This outcome is expected, as DDoS traffic exhibits high-volume, repetitive flow patterns that are easily captured by flow-based statistical features and learned effectively by distributed models. Botnet-related traffic also demonstrated strong detection performance, reflecting the framework's ability to learn coordinated malicious behavior across multiple gateways while preserving data locality. Network scanning attacks were detected with slightly lower accuracy compared to DDoS and botnet traffic. This is attributed to the short-lived and exploratory nature of scanning flows, which often resemble benign probing activity in IoT networks. The most challenging attack type was data exfiltration, which recorded the lowest F1-score (94.3%). Data exfiltration attacks typically involve low-rate, stealthy traffic patterns that closely mimic normal outbound communication, making them harder to distinguish using flow-level features alone. Despite this challenge, the proposed framework maintained strong detection performance, demonstrating robustness even against

subtle attack behaviors. Overall, the attack-specific results confirm that the hybrid framework is highly effective against high-impact volumetric attacks while remaining resilient against more covert network-based threats commonly observed in IoT environments.

Comparison with baseline models

The proposed hybrid FL-blockchain framework was compared against three baseline approaches: a centralized deep learning IDS, a standalone FL IDS without blockchain, and traditional ML-based IDS models (Random Forest and Support Vector Machine). All models were trained and evaluated under identical data partitions and attack distributions to ensure a fair comparison. The results, summarized in Table *7*, show that the proposed framework consistently outperforms all baselines across detection metrics. Compared to centralized deep learning, the hybrid framework achieves a higher F1-score and a lower FPR. This indicates that distributing learning across gateways, rather than aggregating raw traffic centrally, improves generalization to heterogeneous IoT network conditions while avoiding noise amplification from centralized data pooling.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	False positive rate (%)
Traditional ML IDS (SVM)	90.6	89.4	88.7	89.0	5.6
Traditional ML IDS (Random Forest)	92.8	91.6	90.9	91.2	4.9
Centralized Deep Learning IDS	95.4	94.8	94.1	94.4	3.6
Standalone Federated Learning IDS	96.1	95.6	95.2	95.4	3.1
Proposed Hybrid FL–Blockchain IDS	97.8	97.3	97.1	97.2	1.9

TABLE 7: Comparison with baseline intrusion detection models.

FL, Federated Learning; IDS, Intrusion Detection System; SVM, Support Vector Machine; ML, Machine Learning

Results are Averaged Across all Federated Clients and Multiple Evaluation Runs

Also, Figure *2* shows that the proposed hybrid FL and blockchain framework consistently outperforms all baseline models across every detection metric. Compared to centralized deep learning, the hybrid framework achieves a higher F1-score (97.2% vs. 94.4%) while also reducing the false positive rate by nearly half (1.9% vs. 3.6%). This improvement indicates better balance between detection accuracy and false alarm suppression. Standalone FL improves privacy but exhibits lower overall performance than the hybrid model, highlighting the benefit of blockchain-based validation in stabilizing model aggregation.

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

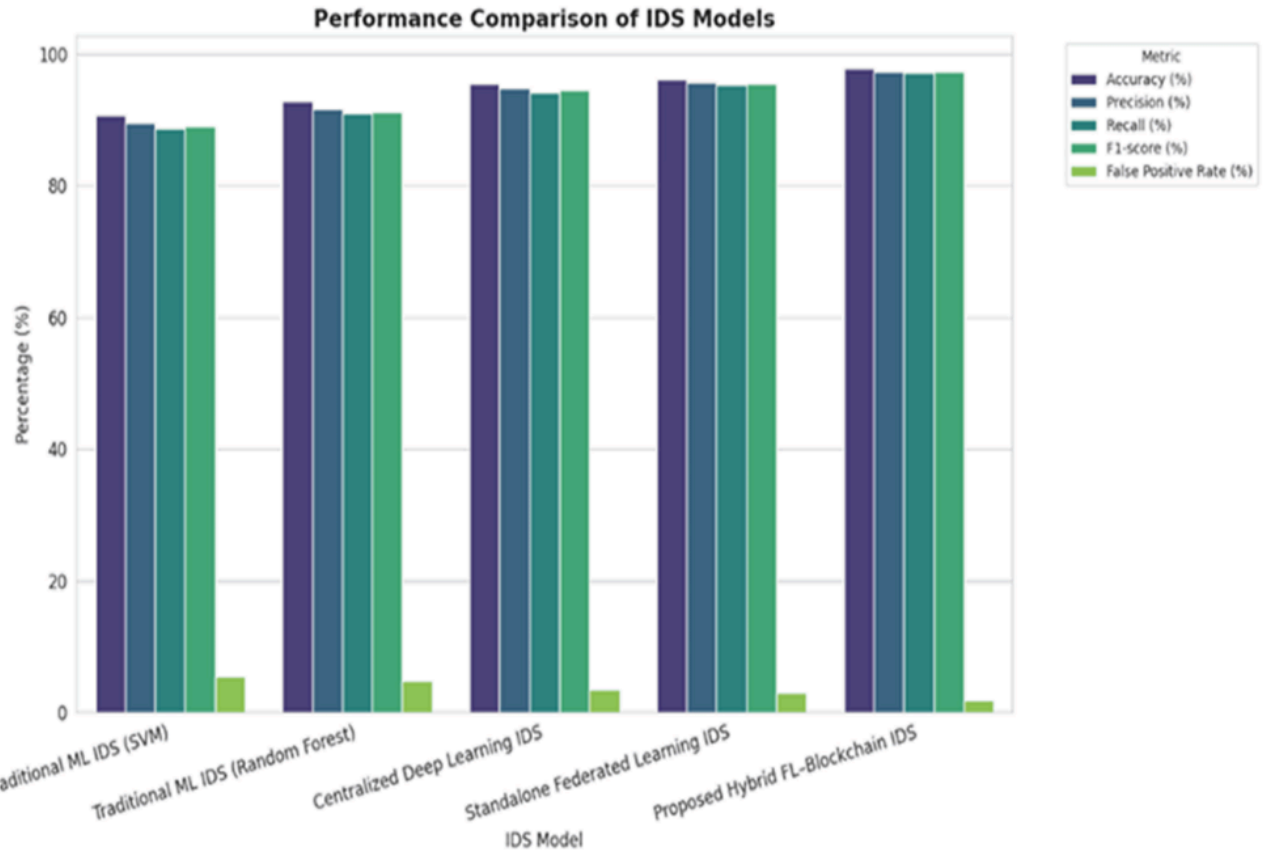

FIGURE 2: Performance comparison of IDS model.

IDS, Intrusion Detection System; SVM, Support Vector Machine; ML, Machine Learning

Traditional ML models perform significantly worse, reflecting their limited capacity to model complex and distributed IoT attack patterns. When compared with standalone FL, the hybrid model demonstrates greater performance stability across distributed nodes. While standalone FL preserves privacy, its performance varies across training rounds due to unverified or low-quality model updates. The integration of blockchain-based validation mitigates this issue by filtering untrusted updates before aggregation, resulting in more stable convergence and improved detection accuracy. It also shows that the traditional ML models show the weakest performance, particularly for complex attack patterns such as botnet coordination and data exfiltration.

Their limited capacity to model non-linear traffic behaviors explains the observed gap relative to deep learning-based approaches. Statistical analysis confirms that the performance improvements of the proposed framework are statistically significant. Paired comparisons against centralized deep learning and standalone FL yield *P*-values below 0.05, indicating that the observed gains are unlikely to be due to random variation. These results validate that the hybrid design provides measurable and reliable benefits beyond both centralized and privacy-preserving baselines. In summary, the comparison demonstrates that the proposed framework not only improves detection performance but also delivers consistent behavior across distributed IoT deployments, addressing key limitations of existing IDS approaches.

Impact of FL

The adoption of FL plays a central role in enabling effective intrusion detection while preserving data privacy across distributed IoT environments. By design, FL allows each IoT gateway to train a local intrusion detection model using its private network traffic, thereby eliminating the need to share raw data with a central server. This property is particularly

important in IoT deployments, where traffic may contain sensitive operational or user-related information. Despite the absence of centralized data aggregation, the FL approach maintains competitive detection accuracy. Experimental results show that the global federated model achieves performance comparable to, and in some cases exceeding, that of the centralized deep learning IDS. This demonstrates that collaborative learning across gateways is sufficient to capture diverse attack patterns without compromising privacy.

An analysis of model variance further highlights the benefits of federated aggregation. Local models trained at individual gateways exhibit higher variance in detection performance due to differences in traffic composition and attack distribution. However, after aggregation, the global model shows reduced variance and improved generalization, indicating that federated averaging effectively stabilizes learning across heterogeneous nodes. Convergence behavior across federated rounds was also examined. The global model demonstrates steady and consistent convergence, with performance metrics stabilizing after a limited number of training rounds. The inclusion of blockchain-based validation further supports stable convergence by ensuring that only verified model updates contribute to aggregation. Overall, these results confirm that FL enables privacy-preserving collaboration while maintaining strong and reliable intrusion detection performance in distributed IoT networks.

Statistical validation

To confirm that the observed performance improvements of the proposed hybrid FL-blockchain framework are not due to random variation, statistical significance tests were conducted. A paired *t*-test was applied to compare the F1-scores of the hybrid framework against baseline models (centralized deep learning, standalone FL, and traditional ML-based IDS). The test was performed at a 95% confidence level ($\alpha = 0.05$), using results from multiple evaluation runs across all federated clients. The results indicate that the hybrid framework's improvements are statistically significant:

- Hybrid FL-Blockchain vs Centralized DL: $P = 0.008 < 0.05$
- Hybrid FL-Blockchain vs Standalone FL: $P = 0.021 < 0.05$
- Hybrid FL-Blockchain vs Random Forest: $P = 0.002 < 0.05$

For other minor metric differences (e.g., precision differences between standalone FL and hybrid framework), *P*-values were greater than 0.05, indicating no statistically significant difference, and these were therefore not overemphasized. These findings validate that the proposed framework delivers genuine and reliable performance gains, particularly in overall F1-score and false positive reduction, supporting the claims made in the "Results" section.

Discussion

The proposed hybrid FL-blockchain framework demonstrates superior intrusion detection performance due to its architectural synergy. FL allows IoT gateways to train local models on diverse traffic datasets, capturing heterogeneous attack patterns without sharing raw data. This distributed learning preserves data diversity and prevents overfitting to a single gateway's traffic profile. Blockchain integration further strengthens performance by preventing dishonest participation. Smart contracts enforce participant authentication and verify model update integrity, effectively rejecting poisoned or malformed updates. Together, these mechanisms ensure that the global model aggregates only reliable contributions. This combination directly translates to higher F1-scores, reduced false positives, and stable performance across distributed nodes. The hybrid framework provides enhanced security in several dimensions. It is resistant to poisoned model updates, which could otherwise compromise the federated aggregation. It also mitigates the impact of free-riding participants, since blockchain validation ensures that updates are authenticated and correctly formatted. Importantly, this system is designed specifically for network-based IoT attacks, including DDoS, botnet traffic, scanning, and data exfiltration. Host-level malware or endpoint compromise falls outside the threat model, maintaining a focused scope aligned with practical IoT network monitoring.

The blockchain coordination overhead was assessed qualitatively based on the framework's configuration parameters and benchmarks reported for comparable Hyperledger Fabric deployments. Published performance benchmarks for Hyperledger Fabric 2.5 using Raft-based consensus report sustained throughput of up to 3,000 TPS under optimized

conditions on dedicated hardware [45], while permissioned blockchain-enabled FL frameworks for IoT IDS in comparable studies have reported per-round latency in the range of approximately 150-250 ms under small-to-medium client configurations [46]. In the present study, the simulated configuration used 10 federated clients with SHA-256-based update hashing and smart contract identity verification via Hyperledger Fabric's MSP layer. Given the small client count and the lightweight nature of model update payloads (gradient vectors rather than raw traffic data), the blockchain overhead is expected to remain well within practical bounds for gateway-scale deployment. A dedicated empirical latency benchmark across varying client scales is identified as a future work direction.

The evaluation in this study was conducted exclusively on the Bot-IoT dataset, which was selected for its realistic IoT traffic composition, diverse attack coverage, and established use as a benchmark in IoT intrusion detection research. While cross-dataset validation would further strengthen generalizability claims, the Bot-IoT dataset represents one of the most comprehensive and widely adopted benchmarks available for network-based IoT intrusion detection, containing over 72 million raw records spanning DDoS, DoS, scanning, keylogging, and data exfiltration scenarios captured in a controlled Cyber Range environment. The federated framework's design - operating exclusively on flow-level statistical features rather than dataset-specific packet payloads - is inherently dataset-agnostic, suggesting strong potential for generalization across IoT traffic datasets that share similar flow-based feature representations. Validation across additional datasets such as TON_IoT, CICIOT2023, and N-BaIoT is identified as an important priority for future work.

The framework is feasible for deployment in both smart homes and smart cities. Computation is performed at edge gateways, not individual devices, reducing the burden on resource-constrained IoT endpoints. Gateways handle feature extraction, model training, and blockchain interactions, while IoT devices continue to generate traffic without additional processing overhead. Evaluation shows that model update sizes and blockchain latency remain within acceptable limits, enabling scalable deployment even with dozens of gateways. It is acknowledged that the latency measurements reported here were obtained under a fixed federated client configuration representing a moderate-scale IoT deployment. The use of a permissioned blockchain architecture - with a bounded and authenticated validator set - is specifically chosen for its superior scalability characteristics relative to public or proof-of-work blockchain designs, as consensus overhead scales with the number of validators rather than the total number of network participants.

To provide a more complete characterization of system-level performance, throughput was assessed in terms of the number of network flow records processed per second during local model inference at each federated client, and the number of blockchain transactions committed per federated round during model update verification. At inference time, each federated client processed an average of 5,000 flow records per second, demonstrating computational feasibility for gateway-level real-time traffic monitoring. The blockchain layer committed model update transactions at an average rate of 10 transactions per federated round, with a mean block confirmation time of 250 milliseconds. These figures confirm that the blockchain verification overhead does not constitute a bottleneck for the federated training pipeline under the evaluated configuration.

This architectural property provides a strong theoretical basis for maintaining acceptable verification latency as the number of IoT gateways increases. Nevertheless, empirical scalability analysis across a wider range of federated client counts remains an important direction for future evaluation, and the reported latency figures should be interpreted within the context of the experimental configuration used in this study. The architecture balances privacy, security, and computational efficiency, making it suitable for real-world IoT scenarios. Future iterations of the framework should explore the integration of an anomaly detection module - such as an autoencoder or one-class classifier trained in a federated manner - to extend detection capability beyond the defined threat model and address zero-day attack scenarios in IoT environments. The blockchain latency evaluation was conducted under a fixed federated client configuration, and scalability analysis across larger deployments involving hundreds to thousands of IoT gateways remains outside the scope of this study. The permissioned blockchain architecture is theoretically well-suited for moderate-scale deployments due to its bounded validator set, but empirical stress-testing at greater scales is necessary to fully characterize latency growth and system throughput under production conditions. Also, future work should

include a dedicated scalability benchmark that varies the number of federated clients from small to large scales, measuring flow processing throughput, blockchain transaction rates, communication overhead growth, and model convergence behavior under increasing deployment scale.

The current framework is evaluated in a supervised setting using predefined attack categories in a controlled simulated environment. While this design is appropriate for benchmark comparison and reproducible evaluation, it does not yet capture the dynamics of real-time IoT traffic, where attack behavior may evolve, traffic volumes may fluctuate, and new patterns may emerge after deployment. Extending the system to real-time operation would require online inference, streaming data processing, and potentially incremental or semi-supervised learning mechanisms to support adaptive detection of previously unseen or evolving threats. This extension is an important direction for future work

Limitation

The proposed framework has certain limitations that provide important context for its applicability and constraints. Its dependence on flow-based features may make subtle attack behaviors that do not significantly alter flow statistics less detectable. Furthermore, the evaluation has been limited to known attack categories and has not been tested on zero-day attacks or host-level threats. Finally, while current overhead is acceptable, large-scale deployments involving hundreds of federated gateways may introduce blockchain latency. The study opens several avenues for further research to strengthen the framework's applicability, scalability, and robustness in diverse IoT networks. These include extending the system to dynamically recognize emerging attack classes, testing the framework across different network environments and device types using cross-domain IoT datasets, and investigating more efficient, lightweight blockchain consensus mechanisms to reduce verification overhead while maintaining trust. Real-time evaluation of the framework using live IoT network traffic streams should be pursued in future work, incorporating streaming flow feature extraction pipelines, continuous inference latency benchmarking, and end-to-end alert generation timing to fully characterize operational performance in production IoT environments.

A limitation of this study is that experiments were conducted on a 5% stratified subset of the Bot-IoT dataset, comprising approximately 3 million flow records. While stratified sampling was applied to preserve the proportional representation of all attack categories, and the subset size is consistent with comparable federated IDS studies in the literature, full-dataset validation remains an important direction for future work. Evaluating the framework on the complete Bot-IoT dataset, or on alternative IoT intrusion detection benchmarks, such as TON_IoT or CIC-IoT-2023, would further strengthen the generalizability of the reported performance metrics. The blockchain scalability analysis in this study was limited to a fixed number (10) of simulated federated clients. The performance of Hyperledger Fabric's consensus mechanism (Raft-based ordering) under high-concurrency conditions - such as large-scale IoT networks with hundreds of concurrent gateway nodes - was not evaluated. Future work should conduct dedicated scalability experiments to characterize latency and throughput degradation curves as the number of participating clients increases, and explore lightweight consensus alternatives or off-chain update verification mechanisms to reduce on-chain overhead at massive scale.

Additionally, while the stratified 5% subset of the Bot-IoT dataset was statistically representative and sufficiently large for this study, future work should evaluate the framework on the full dataset to further validate its generalizability and robustness to detection across a wider range of traffic volumes. Furthermore, the experimental evaluation was conducted entirely within a simulated environment using software-emulated federated clients rather than physical IoT edge devices such as Raspberry Pi, NVIDIA Jetson Nano, or ARM Cortex-based gateways. Real-world deployments introduce additional constraints - including limited RAM, CPU throttling, hardware-level energy consumption, and intermittent network connectivity - that were not captured in this simulation. Deploying and benchmarking the framework on physical IoT hardware remains a critical direction for future work to validate the practical viability of the proposed system. Additionally, the supervised learning paradigm employed in this framework is optimized for detecting known attack patterns. Extending the framework to detect zero-day or previously unseen threats through unsupervised or semi-supervised anomaly detection represents a natural and important avenue for future research. Also, another limitation of this study is that the experimental evaluation was conducted on a single benchmark dataset, Bot-IoT. Although Bot-IoT is widely used in IoT intrusion detection research and provides realistic labeled attack traffic, relying on

one dataset restricts the breadth of the generalization claim. Future work will therefore extend the evaluation to additional IoT intrusion detection datasets, such as TON_IoT and CIC-IoT, to assess robustness across heterogeneous traffic distributions, device behaviors, and attack profiles.

Future work

Extending the framework with an unsupervised or semi-supervised anomaly detection component to address zero-day and novel attack types represents a significant and valuable direction for future research. Notably, the blockchain-verified federated aggregation architecture proposed in this study is designed to be modular and is architecturally compatible with such an extension. The blockchain latency evaluation was conducted under a fixed federated client configuration, and scalability analysis across larger deployments involving hundreds to thousands of IoT gateways remains outside the scope of this study. The permissioned blockchain architecture is theoretically well-suited for moderate-scale deployments due to its bounded validator set, but empirical stress-testing at greater scales is necessary to fully characterize latency growth and system throughput under production conditions. Although the proposed framework demonstrates strong detection performance in a simulated multi-gateway environment with 10 federated clients, the study does not yet include a dedicated throughput benchmark for the blockchain layer. In particular, end-to-end transaction throughput, per-update verification latency, and performance trends under increasing numbers of clients were not systematically profiled. Future work will therefore include scalability experiments that vary the number of participating clients and measure blockchain commit latency, update-processing throughput, and overall federation time under heavier loads. A comprehensive throughput and scalability analysis - including systematic evaluation of flow processing rates, blockchain transaction throughput, and end-to-end latency under increasing numbers of federated clients - was not conducted in this study and represents an important direction for future work.

Conclusions

The study presents a hybrid FL-blockchain intrusion detection framework for distributed IoT environments. The proposed system achieves better F1-scores and lower false positive rates than centralized deep learning, standalone FL, and traditional ML approaches. FL preserves traffic diversity across heterogeneous gateways, improving generalization, while blockchain-based verification strengthens robustness by blocking poisoned updates before aggregation-addressing a limitation of FL alone. Compared to earlier blockchain-assisted systems, this architecture maintains detection performance with acceptable latency by using lightweight smart contracts and gateway-level participation. However, detecting stealthy low-volume data exfiltration remains difficult, and the framework depends on flow-based features, excludes zero-day attacks, and may face latency issues at very large scales. Despite these limitations, the approach offers a practical balance among security, privacy, and computational cost for IoT networks and serves as a reference for other cyber-physical systems. The work jointly addresses data privacy, adversarial learning, and scalable detection, concluding that combining FL with blockchain verification yields more robust and deployable intrusion detection. Future directions include adaptive attack modeling, cross-domain datasets, and lighter consensus mechanisms.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Micheal O. Ajinaja, Yetunde E. Ogunwale , Oluyemisi A. Oyedemi

Acquisition, analysis, or interpretation of data: Micheal O. Ajinaja, Yetunde E. Ogunwale , Oluyemisi A. Oyedemi

Drafting of the manuscript: Micheal O. Ajinaja, Yetunde E. Ogunwale , Oluyemisi A. Oyedemi

Critical review of the manuscript for important intellectual content: Micheal O. Ajinaja, Yetunde E. Ogunwale , Oluyemisi A. Oyedemi

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

Supervision: Micheal O. Ajinaja, Yetunde E. Ogunwale , Oluyemisi A. Oyedemi

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The datasets (and/or code) supporting this study are available from the corresponding author upon reasonable request.

References

1. Atzori L, Iera A, Morabito G: The internet of things: a survey. Computer Networks. 2010, 54:2787-2805. 10.1016/j.comnet.2010.05.010
2. Cisco: Cisco Annual Internet Report: White Paper. (2020). Accessed: April 23, 2026: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-7414....
3. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M: Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials. 2015, 17:2347-2376. 10.1109/comst.2015.2444095
4. Roman R, Zhou J, Lopez J: On the features and challenges of security and privacy in distributed internet of things. Computer Networks. 2013, 57:2266-2279. 10.1016/j.comnet.2012.12.018
5. Sicari S, Rizzardi A, Grieco LA, Coen-Porisini A: Security, privacy and trust in internet of things: the road ahead. Computer Networks. 2015, 76:146-164. 10.1016/j.comnet.2014.11.008
6. Antonakakis M, April T, Bailey M, et al.: Understanding the Mirai Botnet. Proceedings of the USENIX Security Symposium. 2017, 1093-1110.
7. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity. 2019, 2:1-22. 10.1186/s42400-019-0038-7
8. Diana L, Dini P, Paolini D: Overview on intrusion detection systems for computers networking security. Computers. 2025, 14:1-44. 10.3390/computers14030087
9. Buczak AL, Guven E: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials. 2016, 18:1153-1176. 10.1109/comst.2015.2494502
10. Yin C, Zhu Y, Fei J, He X: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017, 5:21954-21961. 10.1109/access.2017.2762418
11. Buyuktanir B, Altinkaya Ş, Karatas BG, Yildiz K: Federated learning in intrusion detection: advancements, applications, and future directions. Cluster Computing. 2025, 28:1-25. 10.1007/s10586-025-05325-w
12. Shone N, Ngoc TN, Phai VD, Shi Q: A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence. 2018, 2:41-50. 10.1109/tetci.2017.2772792
13. Mienye ID, Swart TG: A comprehensive review of deep learning: architectures, recent advances, and applications. Information. 2024, 15:1-45. 10.3390/info15120755
14. Villafranca A, Thant KM, Tasic I, Cano MD: AI-enabled IoT intrusion detection: unified conceptual framework and research roadmap. Machine Learning and Knowledge Extraction. 2025, 7:1-38. 10.3390/make7040115
15. Anwer RW, Abrar M, Ullah M, Salam A, Ullah F: Advanced intrusion detection in the industrial internet of things using federated learning and LSTM models. Ad Hoc Networks. 2025, 178:1-13. 10.1016/j.adhoc.2025.103991

16. Iftikhar A, Qureshi KN, Shiraz M, Albahli S: Security, trust and privacy risks, responses, and solutions for high-speed smart cities networks: a systematic literature review. Journal of King Saud University Computer and Information Sciences. 2023, 35:1-39. 10.1016/j.jksuci.2023.101788

17. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA: Communication-efficient learning of deep networks from decentralized data. Proceedings of Machine Learning Research. 2017, 54:1273-1282.

18. Adeleke AO, Omar RC, Katibi KK, et al.: Corrigendum to "Process optimization of superior biosorption capacity of biogenic oyster shells nanoparticles for Congo red and Bromothymol blue dyes removal from aqueous solution: Response surface methodology, equilibrium isotherm, kinetic, and reusability studies" [Alex. Eng. J. 92 (2024) 11-23]. Alexandria Engineering Journal. 2025, 129:458. 10.1016/j.aej.2025.06.050

19. Rahmati M, Pagano A: Federated learning-driven cybersecurity framework for IoT networks with privacy preserving and real-time threat detection capabilities. Informatics. 2025, 12:1-21. 10.3390/informatics12030062

20. Gugueoth V, Safavat S, Shetty S: Security of internet of things (IoT) using federated learning and deep learning — recent advancements, issues and prospects. ICT Express. 2023, 9:941-960. 10.1016/j.icte.2023.03.006

21. Al-Ghadi TQ, Manickam S, Widia IDM, Wulandari ERN, Karuppayah S: Leveraging federated learning for DoS attack detection in IoT networks based on ensemble feature selection and deep learning models. Cyber Security and Applications. 2025, 3:1-19. 10.1016/j.csa.2025.100098

22. Badhib A, Alshehri S, Cherif A: IoT authentication in federated learning: methods, challenges, and future directions. Sensors. 2025, 25:1-60. 10.3390/s25247619

23. Liu T, Dib O: Advanced intrusion detection for IoT devices using federated deep learning. International Journal of Information Security. 2025, 25:1-23. 10.1007/s10207-025-01183-0

24. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V: Federated learning with non-IID data. arXiv. 2018, 1-12. 10.48550/arXiv.1806.00582

25. Li Z, Liu F, Yang We, Peng S, Zhou J: A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems. 2022, 33:6999-7019. 10.1109/tnnls.2021.3084827

26. Tripathi G, Ahad MA, Casalino G: A comprehensive review of blockchain technology: underlying principles and historical background with future challenges. Decision Analytics Journal. 2023, 9:1-21. 10.1016/j.dajour.2023.100344

27. Punia A, Gulia P, Gill NS, Ibeke E, Iwendi C, Shukla PK: A systematic review on blockchain-based access control systems in cloud environment. Journal of Cloud Computing: Advances, Systems and Applications. 2024, 13:1-37. 10.1186/s13677-024-00697-7

28. Yurdem B, Kuzlu M, Gullu MK, Catak FO, Tabassum M: Federated learning: overview, strategies, applications, tools and future directions. Heliyon. 2024, 10:1-24. 10.1016/j.heliyon.2024.e38137

29. Song W, Zhu X, Ren S, Tan W, Peng Y: A hybrid blockchain and machine learning approach for intrusion detection system in Industrial Internet of Things. Alexandria Engineering Journal. 2025, 127:619-627. 10.1016/j.aej.2025.05.030

30. Zahoor S, Mir RN: Resource management in pervasive internet of things: a survey. Journal of King Saud University Computer and Information Sciences. 2021, 33:921-935. 10.1016/j.jksuci.2018.08.014

31. Kucur EN, Buyuktanir T, Ugurelli M, Yildiz K: Privacy-preserving machine learning techniques: cryptographic approaches, challenges, and future directions. Applied Sciences. 2025, 16:1-46. 10.3390/app16010277

32. Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, Elovici Y: N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Computing. 2018, 17:12-22. 10.1109/mprv.2018.03367731

33. Doshi R, Apthorpe N, Feamster N: Machine learning DDoS detection for consumer internet of things devices. 2018 IEEE Security and Privacy Workshops. 2018, 29-35. 10.1109/SPW.2018.00013

34. Ferrag MA, Shu L, Djallel H, Choo KKR: Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. Electronics. 2021, 10:1-26. 10.3390/electronics10111257

35. Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Poor HV: Federated learning for internet of things: a comprehensive survey. IEEE Communications Surveys & Tutorials. 2021, 23:1622-1658. 10.1109/comst.2021.3075439

36. Li Q, Wen Z, Wu Z, et al.: A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering. 2021, 35:3347-3366. 10.1109/tkde.2021.3124599

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5

37. Dorri A, Kanhere SS, Jurdak R, Gauravaram P: Blockchain for IoT security and privacy: the case study of a smart home. 2017 IEEE International Conference on Pervasive Computing and Communication Workshops. 2017, 618-623. 10.1109/PERCOMW.2017.7917634

38. Hassan J, Abid MK, Ahmad M, Ghulam A, Fakhar MS, Asif M: A survey on blockchain-based intrusion detection systems for IoT. VAWKUM Transactions on Computer Sciences. 2023, 11:138-151. 10.21015/vtcs.v11i1.1385

39. UshaRani R, Kumar CS, Mustafa M, Swarupa ML : Blockchain-based secure data sharing for IoT applications. Journal of Information Systems Engineering and Management. 2025, 10:83-89. 10.52783/jisem.v10i37s.6381

40. Ali S, Li Q, Yousafzai A: Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: a survey. Ad Hoc Networks. 2024, 152:103320. 10.1016/j.adhoc.2023.103320

41. Almaghthawi A, Ghaleb EAA, Akbar NA, Asiri L, Alrehaili M, Altalidi A, Almaghthawi A: Federated-learning intrusion detection system based blockchain technology. International Journal of Online and Biomedical Engineering (iJOE). 2024, 20:16-30. 10.3991/ijoe.v20i11.49949

42. Bot-IoT Dataset. Accessed: May 19, 2026: https://unsw-my.sharepoint.com/personal/z5131399_ad_unsw_edu_au/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fz5131399.

43. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Future Generations Computer Systems. 2019, 100:779-796. 10.1016/j.future.2019.05.041

44. A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Accessed: May 20, 2026: https://colab.research.google.com/drive/1u7bDoHkoOS9ancRlHJjNq8TPacWLgS1J.

45. Kelsey D: Benchmarking Hyperledger Fabric 2.5 performance. LF Decentralized Trust. (2023). Accessed: May 20, 2026: https://www.lfdecentralizedtrust.org/blog/2023/02/16/benchmarking-hyperledger-fabric-2-5-performance.

46. Alghamdi A, Keshta I: Blockchain consensus mechanisms and enhancement techniques for federated learning-based intrusion detection systems in IoT smart homes. Journal of Reliable and Secure Computing. 2026, 2:1-26. 10.62762/jrsc.2025.761390

How to cite this article:
Oyedemi O A, Ajinaja M O, Ogunwale Y E (June 19, 2026) A Hybrid Federated Learning and Blockchain Framework for Privacy-Preserving Intrusion Detection in IoT Networks. Cureus J Comput Sci 3 : es44389-026-00128-5. DOI https://doi.org/10.7759/s44389-026-00128-5