Volume 11, Issue 6, June – 2026 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

Explainable Federated Learning for Secure IoT Networks

Riya Patel[1]; Santosh Saha[1]

[1]Asha M. Tarsadia Institute of CS and T Uka Tarsadia University, Surat, Gujarat, India

Publication Date: 2026/06/26

Abstract: The current increase in cyber threats highlights the need for an advanced intrusion detection system.While IoT networks have be-come more vulnerable to various types of network-level attacks, most AI-powered IDSs are centralized, black-box, and susceptible to manipu-lations of their models' parameters. This work proposes XAI-FedGuard, a federated learning approach which addresses all three issues simultane-ously. The edge devices locally train a lightweight CNN-LSTM classifier based on traffic data, so there is no need to send raw traffic informa-tion to the server. A SHA-256 hash-chain mechanism ensures integrity by checking whether any model updates can possibly be used in poi-soning attacks regardless of their degree of subtlety. Upon global model convergence, KernelSHAP explains every classification decision based on certain network features. We perform experiments on the IoT Vulnerabil-ity Dataset for five attack classes and demonstrate that XAI-FedGuard achieves 97.60% accuracy with an AUC of 0.9978, achieving up to 0.41 percentage point of the upper bound obtained from centralization, while adding merely 3.51 ms per iteration and detecting all instances of attacks on its integrity layer in simulation mode.

Keywords: *IoT Security · Federated Learning · Intrusion Detection · Explainable AI · Model Integrity*.

How to Cite: Riya Patel; Santosh Saha (2026) Explainable Federated Learning for Secure IoT Networks. *International Journal of Innovative Science and Research Technology*, 11(6), 1423-1429. https://doi.org/10.38124/ijisrt/26jun570

I. INTRODUCTION

IoT deployments now span industrial control systems, hospital networks, smart buildings, and consumer devices. The sheer number of endpoints, projected to ex-ceed 30 billion by 2030, makes manual traffic monitoring impossible. Automated intrusion detection is not optional, it is the only realistic defence. The question is what kind of intrusion detection is actually good enough for production use.

Most published IDS models are centralised. Traffic data from every node is shipped to a single server, a model is trained there, and predictions are returned. That works in a lab. In a real deployment it creates two problems. First, raw network traffic often contains sensitive operational data that organisations are not willing to expose to a central collection point. Second, the central server is itself an attractive target: compromise it and the detection system fails en-tirely. Federated learning sidesteps both issues by keeping training data local, but it introduces a new vulnerability: a compromised edge device can now sub-mit deliberately corrupted model updates that quietly degrade global detection accuracy without triggering any alarm.

There is a third problem that cuts across both centralised and federated approaches. Virtually every published IDS model is a black box. It produces a label, attack or benign, without explaining which traffic features drove that decision. For a network administrator trying to write firewall rules or file an incident report, an unexplained label is nearly useless. Regulations in critical infrastructure sectors increasingly require that automated security decisions be auditable, and an AUC score in a paper does not satisfy that requirement.

This paper presents XAI-FedGuard, which handles all three problems in a single architecture. The main contributions are as follows.

- A federated CNN-LSTM training pipeline in which raw IoT traffic data stays on the edge device throughout training.
- A SHA-256 hash-chain integrity protocol that verifies each client's model update before aggregation, with per-round server nonces to prevent replay attacks.
- KernelSHAP attribution computed on the converged global model, providing per-class feature rankings that translate model decisions into terms a security analyst can act on.
- Empirical evaluation on the IoT Vulnerability Dataset against two baselines: centralised CNN-LSTM and standard FedAvg without security extensions.

Section 2 covers related work. Section 3 summarizes the proposed framework. Section 4 describes the experiment results and Section 5 concludes.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

II. RELATED WORK

Machine learning for network intrusion detection has a long history. Classical ap-proaches, random forests, SVMs, k-NN, work well on datasets like NSL-KDD and UNSW-NB15 but struggle with the temporal structure of IoT traffic, where the sequence of packets matters as much as their individual features [1]. CNN-LSTM hybrids address this: the convolutional stage extracts local feature correlations, and the LSTM stage models how those correlations evolve over time [2]. Re-ported detection accuracies on modern datasets routinely exceed 95%, which is why this architecture forms the local model in XAI-FedGuard.

Federated learning for IDS was proposed as a response to the privacy and single-point-of-failure problems of centralised training [3]. FedAvg [4], the stan-dard aggregation algorithm, averages client weights proportional to dataset size. It preserves accuracy well under IID data partitions but is vulnerable to model poisoning: a compromised client can submit weights that nudge the global model toward misclassifying a target class [5]. Several defences have been proposed, including gradient clipping, Krum, and FLTrust, but they typically require sta-tistical assumptions about the distribution of honest updates that are hard to verify in practice. The hash-chain approach in XAI-FedGuard does not make.

Table 1 Comparison of Related IDS Approaches Against XAI-FedGuard Design Criteria.

Reference	FL	Expl.	Integrity	Real IoT	Multi-Class
Bhattacharjee et al. [9]	No	No	No	No	Partial
Reis [10]	Partial	No	No	No	Yes
Latif et al. [5]	Yes	No	Partial	No	No
Fatema et al. [8]	Yes	Partial	No	No	Partial
Dirin et al. [11]	No	No	Partial	Partial	No
XAI-FedGuard (proposed)	Yes	Yes	Yes	Yes	Yes

Such assumptions, it detects tampering through cryptographic mismatch rather than statistical outlier detection.

Explainability in security-critical ML is an active area. SHAP [6] is the dom-inant post-hoc method because it satisfies axiomatic fairness properties that local approximation methods like LIME [7] do not. Recent work has begun inte-grating SHAP into federated pipelines [8], but none of the published federated-explainable IDS frameworks also include model integrity verification. Table 1 makes this gap visible.

III. PROPOSED FRAMEWORK: XAI-FEDGUARD

➢ *Architecture Overview*

XAI-FedGuard has four layers. The IoT Device Layer handles raw packet cap-ture and preliminary feature extraction. The Edge Computation Layer hosts the local CNN-LSTM classifier on each participating node. The Secure Aggregation Layer runs on a trusted server and is responsible for verifying client updates before merging them. The Explainability Layer runs post-convergence on the aggregation server and produces SHAP attributions that are distributed back to operators. Raw training data flows only within the Edge layer, only model weights and their cryptographic digests cross the network.

➢ *Local CNN-LSTM Model*

Each client trains a shared architecture consisting of a Conv1D layer (32 filters, kernel size 3, ReLU), a MaxPooling1D layer, two stacked LSTM layers (32 units each), a dropout layer (rate 0.1), and a softmax output layer sized to the number of attack classes. Input to the network is a tensor of shape (n_features, 1), where n_features is the number of selected traffic features. The local loss function is categorical cross-entropy and the optimiser is Adam with learning rate 10^{-3}. The convolutional stage captures correlations between simultaneously active fea-tures; the LSTM stages learn how those correlations shift across consecutive flow records. The local training objective for client k is:

$$\mathcal{L}_k(\theta_k) = -\frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} \sum_{c=1}^{C} y_{ic} \log \hat{y}_{ic} \tag{1}$$

Where D_k is the local dataset, y_{ic} is the true class indicator, and $\hat{y}_{ic}$ is the predicted probability for class c.

➢ *Integrity-Aware Aggregation*

Before each round, the server issues a unique nonce r^t to all participating clients. After local training, each client computes:

$$h_k^t = \text{SHA-256}\left(\theta_k^t \parallel r^t\right) \tag{2}$$

And transmits both the updated weights θ^t and the digest h^t to the server. The server recomputes the hash from the received weights and compares it to the claimed digest. Any mismatch caused by in-transit corruption, weight tampering, or replay of a previous round's update causes that client's submission to be excluded. Aggregation then proceeds over the verified set K^t:

$$\theta^{t+1} = \sum_{k \in \mathcal{K}^t} \frac{n_k}{n} \theta_k^t \tag{3}$$

Where n_k is the size of client k's training partition and $n = \sum n_k$. The hash chain across rounds ties each round's verification to the previous one, preventing an attacker from reusing a valid digest from an earlier round.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

➢ *SHAP Explainability*

Once the global model converges, KernelSHAP computes feature attributions over a held-out validation sample. For a given input **x** and model f, the SHAP value for feature j is:

$$\phi_j = \sum_{S \subseteq F \setminus \{j\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f(S \cup \{j\}) - f(S) \right] \quad (4)$$

Where F is the full feature set and S ranges over all subsets excluding j. Global feature importance is the mean absolute SHAP value across validation samples. Per-class attributions are computed separately, giving operators a feature ranking for each attack type. This computation runs once at the server after training and does not touch client data.

Table 2 Model Performance on the Held-Out Test Set (19,519 Samples).

Model	Acc.	Prec.	Rec.	F1	AUC	Time (s)
Centralised CNN-LSTM	98.01%	98.12%	98.01%	98.02%	0.9982	259
Standard FedAvg	97.79%	97.92%	97.79%	97.81%	0.9978	900
XAI-FedGuard	**97.60%**	**97.74%**	**97.60%**	**97.62%**	**0.9978**	**920**

IV. EXPERIMENTS

➢ *Setup*

All experiments ran on Google Colab with an NVIDIA Tesla T4 GPU. The dataset is the IoT Vulnerability Dataset [12], published on Mendeley in May 2024. It contains real network traffic captured from physical IoT devices running un-der Kali Linux-generated attacks, stored as PCAP files with derived CSV fea-ture tables. Five attack classes are present: Backdoor, ICMP Redirect, Password Cracking, SQL Injection, and Vulnerability Scan. After dropping columns with more than 50% missing values, 13 traffic features remained. SMOTE was applied to balance classes before splitting, producing 97,595 samples. The split was 64% training, 16% validation, and 20% test, stratified by class.

Three configurations were evaluated. The *centralised CNN-LSTM* trains on the full training set at a single server with no privacy or security extensions. *Standard FedAvg* trains the same model across $K = 5$ clients using standard weight averaging, with no integrity checking and no SHAP. *XAI-FedGuard* adds SHA-256 integrity verification and post-convergence SHAP attribution to the FedAvg setup. All federated experiments used $T = 10$ communication rounds and $E = 5$ local epochs per round.

➢ *Classification Results*

Table 2 reports test-set performance across all three configurations.

XAI-FedGuard sits 0.41 percentage points below the centralised ceiling. That gap is the combined cost of federated training, integrity checking, and all stochas-tic variation across ten rounds. Standard FedAvg accounts for 0.22 of those points, meaning the integrity and explainability extensions together cost less than 0.2 points of accuracy which is within normal run-to-run variance for this architecture. The AUC values for both federated models are identical at 0.9978, suggesting the ranking quality of predictions is unaffected by the security addi-tions.

Per-class results tell a more specific story. Backdoor and ICMP Redirect both reach F1 of 0.99, which reflects the distinctive port and sequence-number pat-terns these attacks produce. SQL Injection is the hardest class: precision is 0.92 against a perfect recall of 1.00, meaning the model catches every genuine injec-tion attempt but flags some benign traffic as suspicious. Password Cracking and Vulnerability Scan sit between these extremes at F1 of 0.97 and 0.98 respectively.

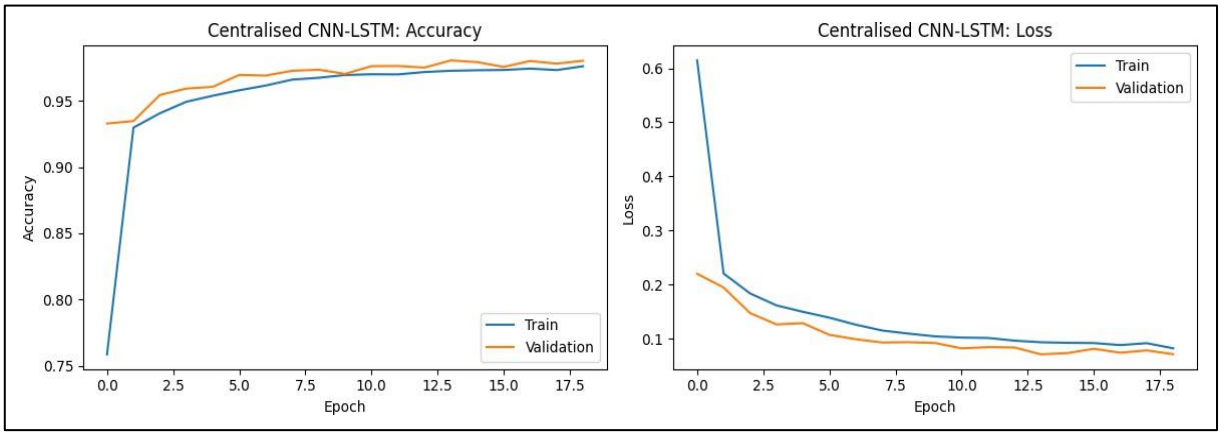

Fig 1 Training and Validation Accuracy/Loss Curves for the Centralised CNN-LSTM Baseline Over 19 Epochs.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

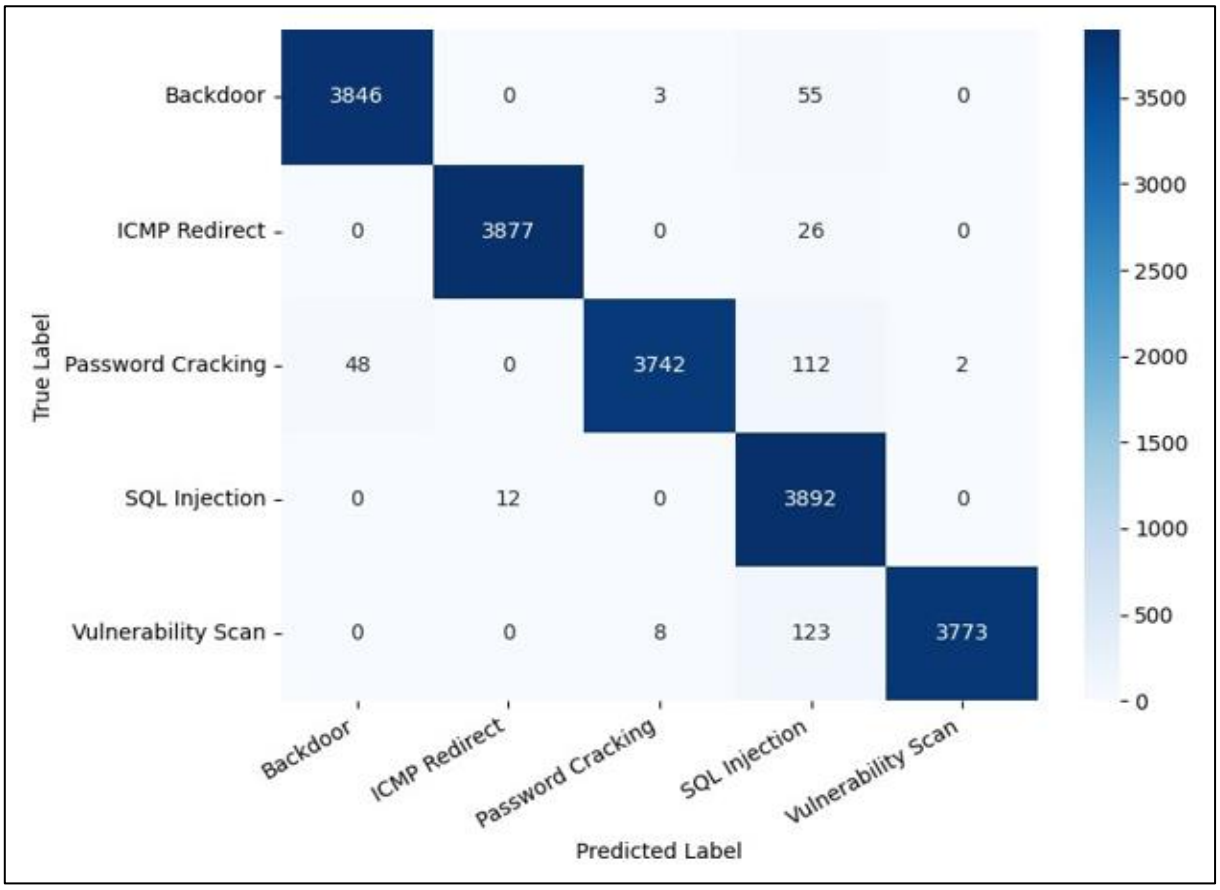

Fig 2 Confusion Matrix for the Centralised CNN-LSTM Baseline on the Five-Class Test Set (19,519 Samples).

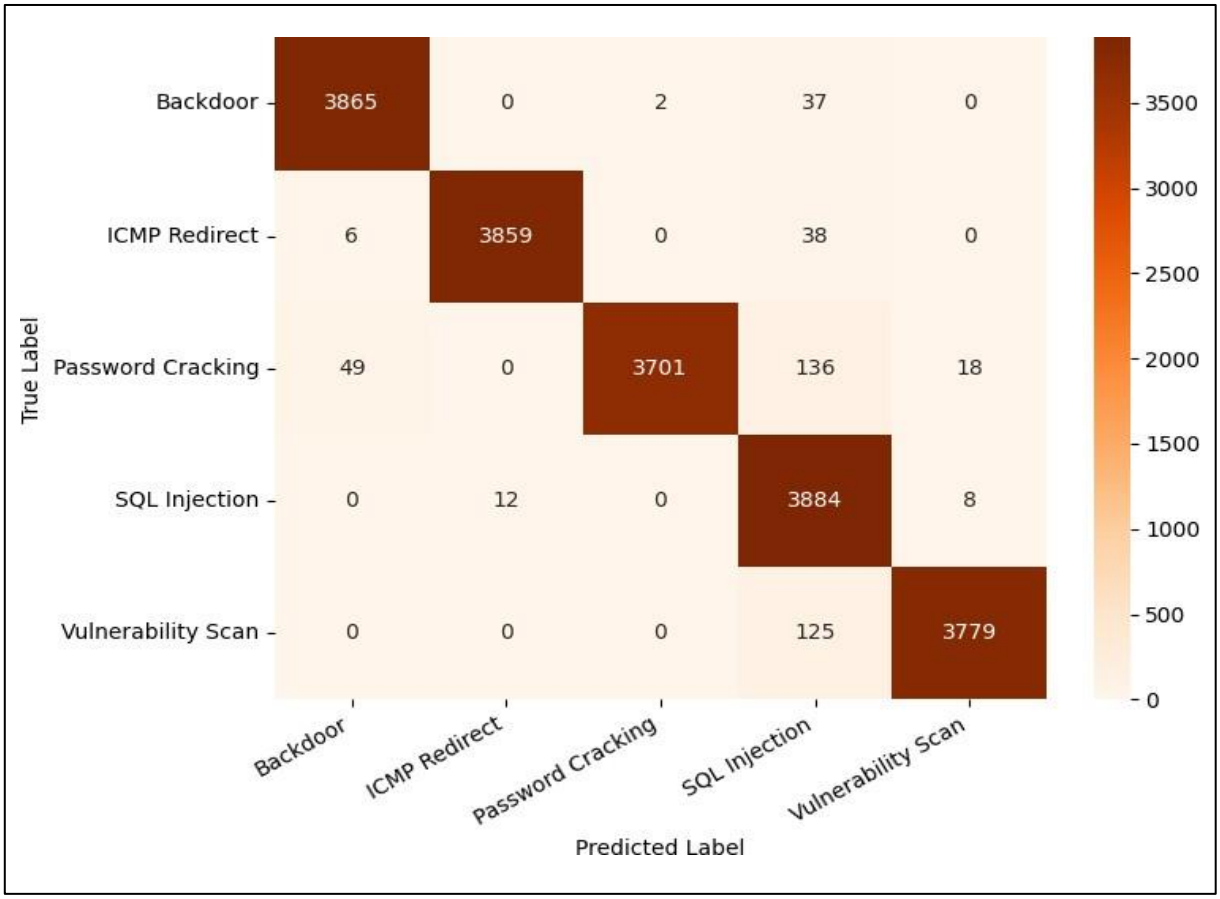

Fig 3 Confusion Matrix for Standard FedAvg on the Five-Class Test Set (19,519 Sam-Ples).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

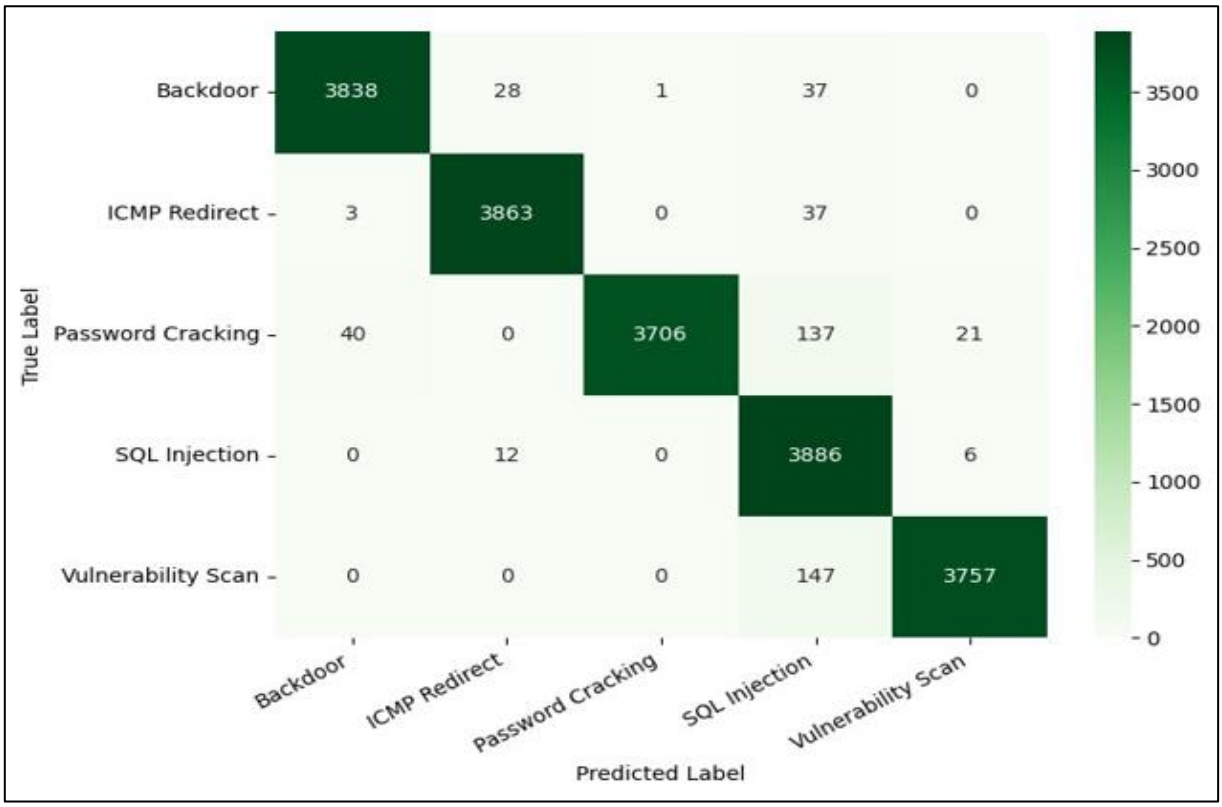

Fig 4 Confusion Matrix for XAI-FedGuard (Proposed) on the Five-Class Test Set (19,519 Samples).

Table 3 Top Discriminative Feature Per Attack Class (KernelSHAP).

Attack Class	Top Feature	Mean $\|\phi\|$	Security Interpretation
Backdoor	Packet No.	0.090	Repetitive, ordered connection sequences
ICMP Redirect	Source Port	0.110	Atypical ICMP source port behaviour
Password Cracking	Packet No.	0.242	High-frequency iterative auth attempts
SQL Injection	Packet No.	0.170	Structured query repetition patterns
Vulnerability Scan	Source Port	0.125	Sequential port sweep across services

➢ *Integrity Verification*

SHA-256 verification averaged 3.51 ms per round across all five clients. Total training time for XAI-FedGuard was 919.81 seconds, compared to 900.29 s for standard FedAvg, a difference of under 20 seconds across ten rounds. As a frac-tion of per-round computation, the integrity overhead is below 0.004%.

To test detection capability, Client 2 was designated as a compromised node across five simulation rounds. In each round, the attacker computed an honest SHA-256 digest before applying large Gaussian noise ($\sigma = 50$) to the weights, then sent the corrupted weights alongside the pre-computed honest hash. The server detected the mismatch in all five rounds, giving a detection rate of 100%. This is the expected result: SHA-256 is collision-resistant, so any change to the weight tensor produces a digest that is computationally infeasible to match against the original, regardless of how targeted the perturbation is.

➢ *SHAP Attribution*

The background size of 100 was selected empirically to balance attribution sta-bility and computational cos, preliminary trials with 50 and 200 background samples yielded consistent top-feature rankings, suggesting the reported attri-butions are robust to this hyperparameter within the tested range. Table 3 shows the top-ranked feature for each attack class along with its mean absolute SHAP value.

Source Port and Packet No. are the dominant discriminators across all five classes. This makes sense: attacks tend to generate traffic with systematic port sequences or high-rate repetition that normal flows do not exhibit. The Acknowl-edgement Number (Ack No RAW) emerges as a secondary discriminator unique to Password Cracking, reflecting the teardown-and-reconnect pattern of repeated login attempts, complementing Packet No. as the dominant feature for this class. TCP Window size is informative for Backdoor detection ($|\phi| = 0.083$), consistent with covert channels that use atypical window negotiations to avoid triggering signature-based rules.

These attributions give a security analyst something to work with beyond a prediction label. If the model flags a flow as a Backdoor attempt, the analyst can check whether the window size and packet sequence number match the pattern the model learned. That is genuinely useful for triage and for writing detection rules.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

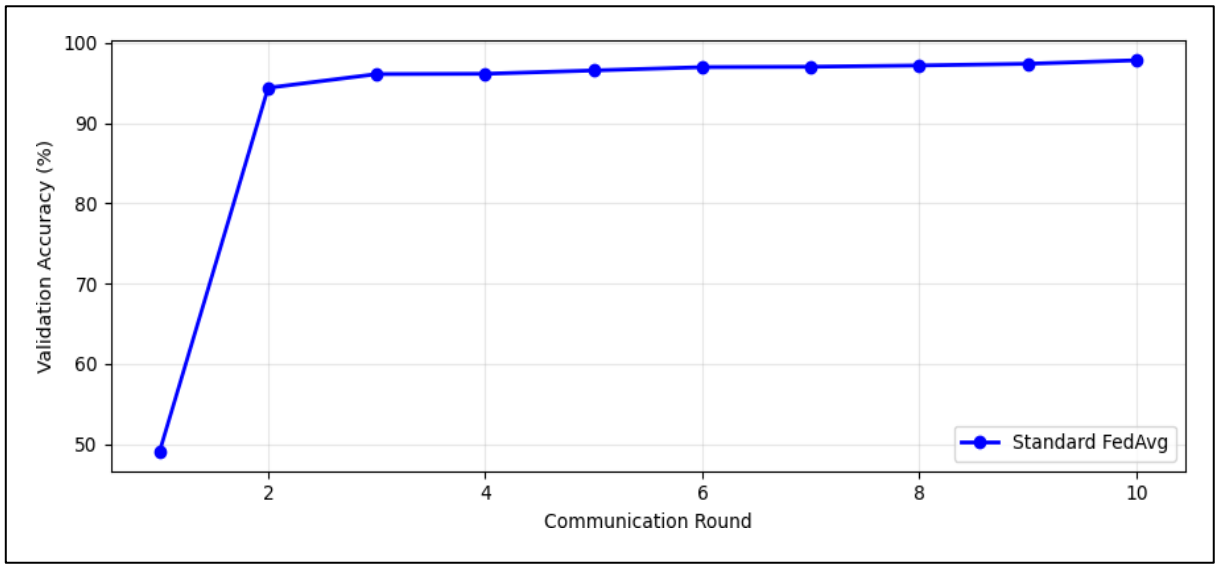

Fig 5 Validation Accuracy Per Communication Round for Standard FedAvg.

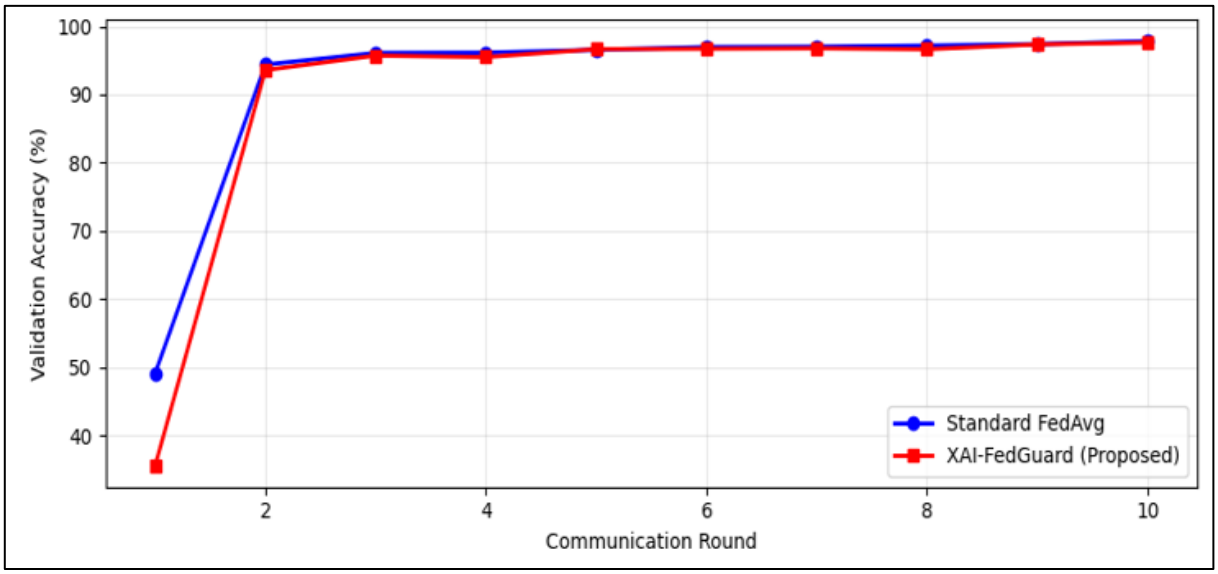

Fig 6 Convergence Comparison Between Standard FedAvg and XAI-FedGuard Across 10 Communication Rounds.

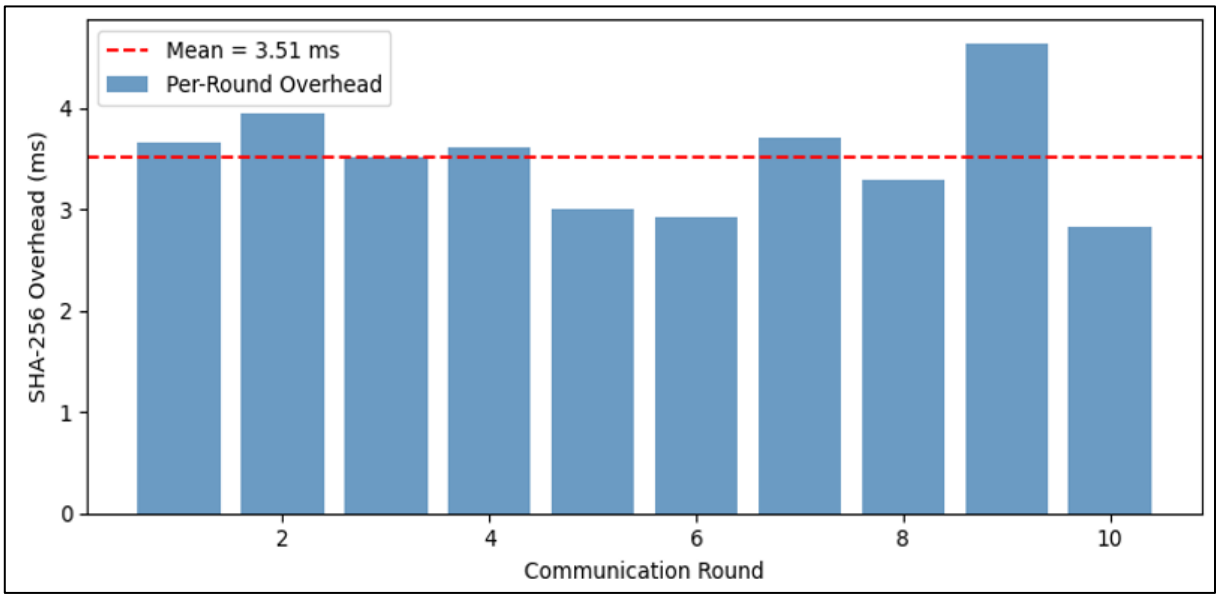

Fig 7 SHA-256 Integrity Verification Overhead Per Communication Round

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun570

V. CONCLUSION

The framework XAI-FedGuard combines three elements that are addressed sep-arately by existing IoT IDS frameworks. First, federated learning that maintains the confidentiality of the raw traffic data, second, the cryptographic verification of the model updates that prevents model poisoning at the aggregation step, and third SHAP attribution that enables model auditability. The model trained on the IoT vulnerability dataset with five attacks achieved 97.60% accuracy and 0.9978 AUC, at the expense of only 3.51 ms of integrity overhead per round, which compares favourably with the 98.01% centralized ceiling, with negligible overhead. However, there are two limitations of the presented results. First, the assumption about IID data partition. Indeed, in the real-world IoT setting it is highly unlikely that the smart-meter fleet will generate data identical to a hospital monitors network, thus causing the non-IID data distribution in the federated model. The way to mitigate this problem is to use personalized feder-ated learning or the approach based on clustered aggregation. Another limitation is the computational complexity of SHAP calculation, as the current 289 seconds of its computation are reasonable in case of the post-learning analysis, but un-acceptable in the case of frequent retraining. Future research will also examine how to combine integrity checks of software components outlined in this paper with hardware attestation of devices, which would ensure an end-to-end chain of trust starting from raw sensors' output and ending at the aggregate model. This would resolve the only gap that currently exists between what XAI-FedGuard is validating and what production deployment requires.

REFERENCES

[1]. Saheed, Y.K., et al.: Machine learning-based intrusion detection for IoT network attacks. Alexandria Engineering Journal 61, 9395–9409 (2022)
[2]. Karimullah, K., et al.: A hybrid CNN-LSTM-based intrusion detection system trained on UNSW-NB15. Journal of Computing and Biomedical Informatics 10(1) (2025)
[3]. Mallidi, S.K.R., Ramisetty, R.R.: Advancements in AI-based intrusion detection systems in IoT: a review. Discover Internet of Things 5, 8 (2025)
[4]. McMahan, B., et al.: Communication-efficient learning of deep networks from de-centralized data. In: Proceedings of AISTATS (2017)
[5]. Latif, S., et al.: Lightweight integrity-driven federated learning for IoT security. IEEE Internet of Things Journal (2024)
[6]. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
[7]. Ribeiro, M.T., et al.: 'Why should I trust you?': explaining the predictions of any classifier. In: Proceedings of KDD (2016)
[8]. Fatema, N., et al.: FedXAI-IDS: federated explainable intrusion detection for IoT. IEEE Access (2025)
[9]. Bhattacharjee, S., et al.: A probabilistic trust framework for secure IoT transmis-sions. IEEE Sensors Journal (2017)
[10]. Reis, A.: Hybrid deep learning for anomaly detection in 5G-enabled smart city IoT. Sensors 22(21), 8417 (2022)
[11]. Dirin, A., et al.: IoTAttest: TPM 2.0 remote attestation for IoT device identity. IEEE Access (2023)
[12]. Koppula, V., Leo Joseph, J.: IoT Vulnerability Dataset. Mendeley Data v1 (2024). https://doi.org/10.17632/7m58kxs742.1
[13]. Alrayes, F.S., et al.: Optimizing intrusion detection using hybrid random forest and CNN-LSTM. Journal of AI Research (2025)
[14]. Bhavsar, M., et al.: Anomaly-based intrusion detection system for IoT applications. Discover IoT 3, 5 (2023)
[15]. Alkhonaini, M.A., et al.: A two-phase spatiotemporal chaos-based protocol for IoT data integrity. Scientific Reports 14(1) (2024)
[16]. Hang, I., Kim, D.: A study of IoT security and blockchain. Sensors 19(7), 1729 (2019)
[17]. Zhao, G., et al.: Privacy-preserving blockchain-based integrity checking for IoT. IEEE Transactions on Cloud Computing (2020)
[18]. Aman, M.N., et al.: Low power data integrity in IoT systems. IEEE Internet of Things Journal 5(4), 3102–3113 (2018)
[19]. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of FedAvg on non-IID data. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=HJxNAnVtDS