A Privacy-Preserving and Interpretable Federated IDS for IoT-Based Networks

Talluru Venkata Jaswanth Chowdary, Settipalli Mahivardhan, Sunkari Ravindra,
Jonnada Praisee Surya Raj, Kiran Gopal Pedireddy, Takellapati Karthikeya, K Nimmy
Department of Computer Science and Technology
Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amritapuri,India

***Abstract*—We propose XFedIDS+, a federated intrusion detection system addressing critical challenges in IoT security: data privacy, device heterogeneity, and computational constraints. The framework employs a CNN-LSTM architecture with attention mechanisms optimized for edge deployment, while FedProx ensures stable training across non-IID data distributions. Our novel CRADS algorithm dynamically selects participating clients based on real-time resource availability (CPU, memory, battery, network), significantly improving training reliability. Integrated SHAP and LIME explainability modules provide interpretable threat analysis for security analysts. Experimental validation on UNSW-NB15 and IoT-23 datasets achieves 93.53% and 96.71% accuracy respectively in federated settings, with successful Raspberry Pi 4 deployment demonstrating 1.66ms inference latency, confirming practical applicability for real-world IoT security.**

***Index Terms*—Federated Learning, Intrusion Detection Systems, Internet of Things (IoT) Security, Explainable Artificial Intelligence (XAI), Resource-Constrained Devices**

I. Introduction

The Internet of Things (IoT) has transformed modern computing by enabling billions of devices to collect, transmit, and analyze data across domains such as healthcare, transportation, and industrial automation. However, this large-scale connectivity introduces a broad attack surface for cyber threats. IoT devices often operate with limited resources and communicate over heterogeneous networks, making them difficult to secure with traditional centralized intrusion detection systems (IDS) [12]. These systems typically require the transmission of raw data to central servers, which raises privacy concerns, creates bandwidth bottlenecks, and makes them unsuitable for latency-sensitive environments [5]. Moreover, their reliance on black-box machine learning models hampers explainability and trust factors that are critical for deployment in security-sensitive applications [4].

To address these limitations, we propose XFedIDS+, a privacy-preserving and interpretable federated learning-based IDS for IoT networks. Federated Learning (FL) enables decentralized model training by sharing model updates instead of raw data, thereby preserving user privacy and reducing communication overhead [1]. XFedIDS+ employs a lightweight CNN-LSTM architecture with attention mechanisms to capture both spatial and temporal network traffic features [2]. The model is optimized for non-IID data and heterogeneous device capabilities through the FedProx algorithm, which stabilizes local training by regularizing updates toward the global model [2]. Unlike lightweight architectures such as MobileNet which often sacrifice temporal learning capabilities, our design balances accuracy and efficiency for edge deployment [13].

To ensure reliability in real-world deployments, XFedIDS+ integrates CRADS (Client Resource-Aware Dynamic Selection), a client selection mechanism based on CPU, memory, battery, and network status [7]. This strategy reduces training failures and enhances model convergence in heterogeneous IoT environments. Furthermore, the framework incorporates explainable artificial intelligence (XAI) through SHAP and LIME, enabling both global and local interpretability of model decisions [3]. Existing FL-based IDS solutions often lack such resource-awareness and interpretability, which limits their practical deployment [8].

We validate XFedIDS+ using two publicly available datasets—UNSW-NB15 [9] and IoT-23 [6]—which encompass diverse attack types and network behaviors. Our system achieves over 92% detection accuracy in federated settings and demonstrates low-latency inference when deployed on a Raspberry Pi 4 device [11]. Through its combination of resource-aware client selection, efficient deep learning architecture, and explainability, XFedIDS+ provides a robust and scalable solution for real-world IoT intrusion detection [10].

Lastly, XFedIDS+ emphasizes practical deployment and transparency—two often-overlooked aspects in federated security research [11]. Its ability to operate on constrained devices while delivering interpretable decisions make it suitable for smart cities, industrial IoT, and critical infrastructure protection [5]. The modular design further allows future integration with decentralized FL architectures and formal privacy-preserving mechanisms [1].

II. Related Work

While Intrusion Detection Systems have made impressive strides with the introduction of machine learning and deep learning techniques [6], adapting these advances to real-world IoT environments has proven to be a complex challenge. The main obstacles preventing widespread adoption include protecting user privacy when devices handle sensitive data, working within the strict computational limits of IoT hardware [5], and providing clear explanations for security decisions that analysts can understand and trust [4].

Federated Learning enables machine learning model training without exposing sensitive data by keeping information on local devices. McMahan et al.'s FedAvg algorithm demonstrates that combining model updates from distributed devices can achieve performance comparable to centralized approaches [1]. However, FedAvg struggles in IoT environments where devices have heterogeneous data distributions and varying capabilities. FedProx addresses these challenges through regularization techniques that maintain stable training across diverse client configurations [2].

Several FL-based intrusion detection systems have been developed to enhance privacy and scalability in IoT networks [6]. However, most existing systems assume uniform device capabilities and overlook practical deployment issues such as limited computational resources or unreliable connectivity [11]. Additionally, few solutions sufficiently address the need for explainable and trustworthy federated model decisions to support human-in-the-loop security workflows [8].

Deep learning models like CNNs and LSTMs are effective at detecting complex attack patterns in network traffic [6], but often demand significant computational power that exceeds the capacity of IoT edge devices. Lightweight alternatives such as MobileNet improve resource efficiency but may compromise detection accuracy or temporal understanding [10]. Hybrid CNN-LSTM architectures, like the one used in XFedIDS+, aim to strike a balance between accuracy and efficiency by learning both spatial and temporal traffic features [2]. However, most evaluations of such models are limited to centralized settings, with few addressing deployment on constrained devices like Raspberry Pi in federated environments [11].

Explainable AI helps security analysts trust automated intrusion detection outputs by clarifying why a system flags certain behaviors. SHAP (Shapley Additive Explanations) identifies globally significant features across all predictions [3], while LIME (Local Interpretable Model-Agnostic Explanations) focuses on understanding individual alerts [4]. While some recent works apply explainability in federated IDS settings [8], real-time explanations on constrained, time-sensitive platforms remain limited.

Federated intrusion detection systems face three core challenges. First, device diversity: most FL-IDS assume uniform computing environments, while actual IoT networks span high-end and low-power devices [6]. Second, lack of explainability: many existing systems fail to provide human-understandable reasons for model predictions, making them difficult to trust in practice [4]. Third, the deployment gap: most studies are based on simulations, not actual field-tested edge devices, which introduces uncertainty in practical viability [11].

Our proposed XFedIDS+ framework addresses these gaps through CRADS-based client selection, a hybrid CNN-LSTM architecture enhanced with attention for pattern detection, and FedProx for stable training in heterogeneous environments [2]. Additionally, it offers explainability using SHAP and LIME, supporting both global system analysis and local decision justification [3]. We validate the framework on UNSW-NB15 and IoT-23 datasets [9], and demonstrate real-world deployability using Raspberry Pi 4, confirming its feasibility for lightweight, edge-level IoT security [11].

III. Proposed System: XFedIDS+

To tackle the challenges of privacy protection, device diversity, and understandable threat detection in IoT networks, we introduce XFedIDS+ ,a federated learning framework that provides secure, scalable, and explainable intrusion detection. Our system consists of four main components: (1) an efficient deep learning model that runs on resource-limited devices, (2) federated optimization using FedProx for stable training across diverse devices, (3) a smart client selection strategy (CRADS) that matches tasks to device capabilities, and (4) dual explanation capabilities that help security analysts understand both overall system behavior and individual threat alerts.

A. System Architecture

The XFedIDS+ framework connects a central coordination server with multiple IoT edge devices throughout the network, as illustrated in Figure 1. Each device trains its own security model using local network traffic and periodically shares model improvements with the central server, which combines these updates to enhance overall threat detection capability without exposing actual data. Each IoT client runs a lightweight CNN-LSTM model that analyzes network traffic in real time and detects suspicious activities using minimal computing resources. The federated server acts as the coordination hub, receiving encrypted model updates from all devices and using the FedProx algorithm to combine them into an improved global security model. The explainability engine provides dual analysis capabilities: SHAP analysis identifies which network features are most important across all

devices, while LIME explains why specific traffic was flagged as threatening. Throughout this process, secure communication ensures all model updates are encrypted during transmission, guaranteeing that sensitive network traffic data never leaves individual devices.

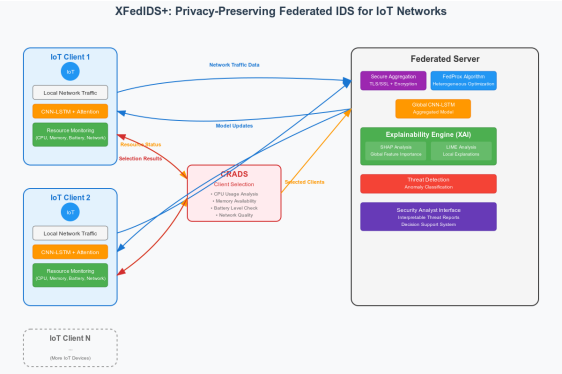

Fig. 1: XFedIDS+ system architecture showing federated workflow and key modules.

To capture both immediate patterns and time-based trends in network traffic, XFedIDS+ uses a hybrid deep learning model combining three complementary techniques. 1D Convolutional layers identify distinctive patterns within individual network packets, similar to image recognition detecting specific shapes. LSTM layers track traffic behavior changes over time, learning to recognize attack sequences that unfold across multiple interactions [6]. The attention mechanism automatically focuses on the most suspicious traffic parts, helping prioritize features critical for threat detection. This architecture balances detection accuracy with computational efficiency, enabling effective operation on resource-constrained IoT devices while capturing complex attack patterns [10].

Traditional federated learning algorithms like FedAvg have difficulty reaching stable solutions when devices have very different types of network traffic data. To solve this problem, XFedIDS+ uses the FedProx algorithm, which keeps individual device models from drifting too far from the shared global model by adding a stabilizing constraint to each device's training process.

$$\min_{w} f_k(w) + \frac{\mu}{2}\|w - w_t\|^2 \tag{1}$$

where $f_k(w)$ is the loss function on client k, w_t is the global model from the previous round, and μ controls the degree of regularization. This encourages local updates to remain close to the global model, improving convergence in heterogeneous environments.

Device availability poses a major challenge in federated learning for IoT environments, where devices often struggle with limited battery life, processing power, and connectivity. Many devices may become unavailable mid-training, disrupting collaborative learning. While existing approaches propose client selection mechanisms to address unreliable clients [7], they often rely on static heuristics or assume availability feedback, which may not reflect real-time resource conditions in IoT devices.

We introduce CRADS, a smart client selection strategy that evaluates device readiness based on four key factors: CPU usage, available memory, battery power, and network quality. Each device is assigned a capability score reflecting its ability to handle local training workloads without failure. During each communication round, the system selects the top-K clients with the highest capability scores for participation. This significantly reduces training failures and improves the reliability and efficiency of the federated learning process.

Algorithm **: CRADS — Client Resource-Aware Dynamic Selection**

Require: List of clients $\mathcal{C}$, threshold θ
Ensure: Selected clients $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$
for each client $c_i \in \mathcal{C}$ **do**
  Obtain resource metrics: $cpu_i, mem_i, batt_i, net_i$
  Compute capability score: $R_i \leftarrow \alpha \cdot cpu_i + \beta \cdot mem_i + \gamma \cdot batt_i + \delta \cdot net_i$
  if $R_i \geq \theta$ **then**
    Add c_i to $\mathcal{S}$
  end if
end for
return $\mathcal{S}$

XFedIDS+ also employs a secure aggregation protocol to protect model updates during transmission. Each client adds a randomized scrambling layer to its local model updates before sharing them with the central server. This ensures that the server can only observe the aggregated results from all participating clients, without inferring any individual device's contribution [5].

$$\sum_{i=1}^{N}(x_i + r_i) - \sum_{j=1}^{N} r_j = \sum_{i=1}^{N} x_i \tag{2}$$

This technique ensures that even if the server is compromised, individual client updates remain confidential. All communications are protected using TLS/SSL encryption.

To help security analysts understand and trust system decisions, XFedIDS+ includes two complementary explanation tools. SHAP provides global insights by showing which network features (packet size, connection frequency, protocol types) are most important for threat detection across all devices and alerts [3]. LIME explains specific alerts by breaking down why particular network traffic was flagged as suspicious, creating simplified explanations for individual cases [4]. These tools enable security analysts to understand and verify the CNN-LSTM

model's threat detection decisions at both network-wide and individual alert levels.

IV. Experimental Setup

We design a comprehensive testing strategy to evaluate how well the proposed XFedIDS+ framework performs in terms of accuracy, interpretability, and real-world applicability. The experiments integrate computational simulations with physical device deployments, using two publicly available cybersecurity datasets commonly used for intrusion detection research.

A. Datasets

We evaluate XFedIDS+ using two diverse cybersecurity datasets that represent different network environments: UNSW-NB15 and IoT-23.

The UNSW-NB15 dataset, released by the Australian Centre for Cyber Security, includes extensive samples of both normal and malicious network activity. It contains nine categories of attacks and 49 extracted network features derived from real traffic, covering threats such as exploits, worms, backdoors, and denial-of-service (DoS) attacks.

The IoT-23 dataset, curated by the Stratosphere Laboratory, focuses on smart device environments. It provides labeled network traffic from actual IoT devices under attack, including botnet activity from Mirai and Gafgyt malware families. The dataset also includes normal device behavior and organizes traffic by malware type and device context, making it well-suited for simulating federated learning scenarios where each client represents a unique IoT environment. This mirrors real-world smart homes and industrial IoT deployments where device behaviors and threat profiles vary.

B. Data Preprocessing

For both datasets, we apply comprehensive preprocessing steps to prepare the data for federated learning. Feature selection removes non-numeric identifiers such as IP addresses and port numbers to focus on behavioral patterns. Label encoding converts categorical labels into binary classes of benign (0) and malicious (1) traffic. Min-Max normalization scales all feature values to the range [0, 1] for consistent model training. Finally, federated partitioning splits the datasets into multiple non-IID partitions to simulate heterogeneous clients, with each client receiving traffic data from distinct device subsets or malware families, reflecting realistic IoT deployment scenarios.

C. Federated Learning Simulation

We simulate a federated environment with $K = 5$ clients, each having local data and computational resources. The training configuration used a fixed number of global communication rounds with 5–10 local epochs per client per round. Client selection compares random versus CRADS-based resource-aware strategies, while model aggregation employed FedProx with tunable μ parameter. Training uses the Adam optimizer with varying batch sizes across clients to simulate realistic heterogeneity in IoT deployments.

D. Hardware and Software Setup

The experiments are conducted on a central server with an AMD Ryzen 7 processor, 16 GB RAM, and Ubuntu 22.04, while clients are simulated using isolated Python environments, with a Raspberry Pi used for deployment validation. The software stack includes TensorFlow 2.x for training and TFLite conversion, Python 3.10, SHAP, and LIME for explainability analysis, and Matplotlib/Seaborn for result visualization.

E. Evaluation Metrics

To comprehensively assess model performance and deployment suitability, we employ multiple evaluation metrics. Classification performance is measured using accuracy (fraction of correct predictions), precision, recall, and F1-score for imbalanced datasets, along with false positive rate (FPR) and false negative rate (FNR) as critical indicators for intrusion detection effectiveness. Deployment viability was evaluated through communication overhead (total volume of model updates per round), inference latency on Raspberry Pi devices, and model size and memory usage after TFLite quantization for edge deployment optimization.

V. Results and Discussion

To evaluate the performance and interpretability of the proposed XFedIDS+ framework, we conducted experiments on the UNSW-NB15 and IoT-23 datasets. Each subsection compares centralized vs. federated performance, provides SHAP and LIME-based interpretability, and analyzes CRADS vs. accuracy across rounds.

A. UNSW-NB15 Evaluation

TABLE I: UNSW-NB15: Centralized vs. Federated (FedProx and FedAvg) + CRADS

Mode	Acc.	Prec.	Recall	F1	FPR	FNR
Centralized	97.00%	95.00%	96.00%	96.00%	0.53%	2.99%
FedAvg + CRADS	89.22%	90.72%	92.61%	91.65%	16.78%	7.39%
FedProx + CRADS	93.53%	98.45%	91.31%	94.75%	2.55%	8.69%

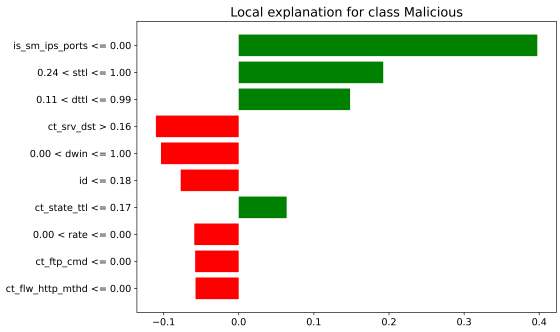

Fig. 2: LIME explanation for a sample in UNSW-NB15

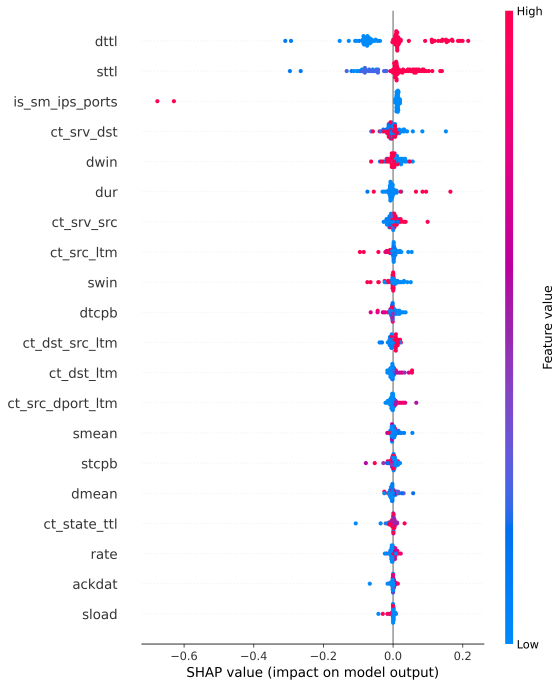

Fig. 3: SHAP feature importance for UNSW-NB15

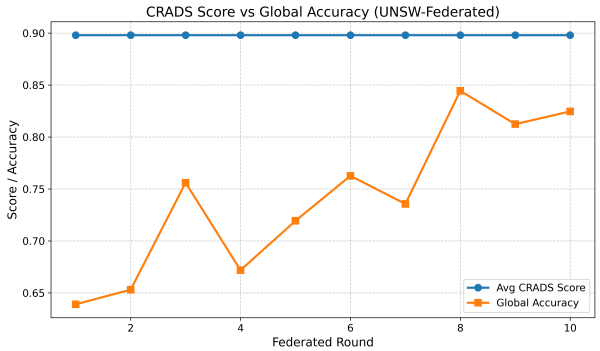

Fig. 4: CRADS Score vs. Global Accuracy (UNSW-Federated)

Centralized training achieves higher overall performance due to complete data access, while federated learning experiences slight accuracy drops from client heterogeneity and non-IID distributions. However, the federated model maintains strong precision and recall, making it viable for privacy-preserving intrusion detection. SHAP analysis reveals that features like `total_bytes` and `bytes_per_pkt` significantly influence model decisions, validating learned patterns. The CRADS client selection mechanism improves global model performance by prioritizing clients with higher resource availability, enhancing training stability in federated settings.

B. IoT-23 Evaluation

TABLE II: IoT-23: Centralized vs. Federated + CRADS

Mode	Acc.	Prec.	Recall	F1	FPR	FNR
Centralized	95.73%	95.35%	96.16%	95.75%	4.69%	3.84%
FedAvg + CRADS	87.11%	80.98%	97.00%	88.27%	22.78%	3%
FedProx + CRADS	96.71%	94.97%	98.66%	96.78%	5.23%	1.34%

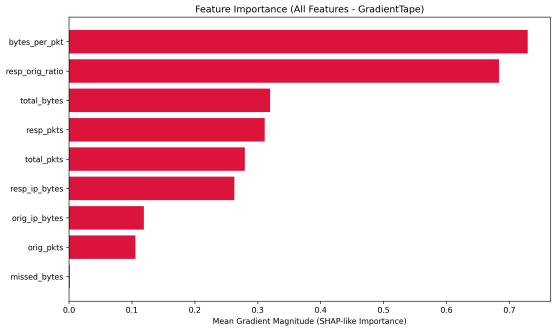

Fig. 5: SHAP feature importance for IoT-23

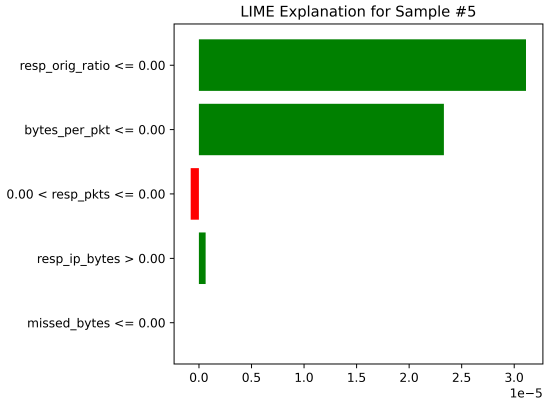

Fig. 6: LIME explanation for a sample in IoT-23

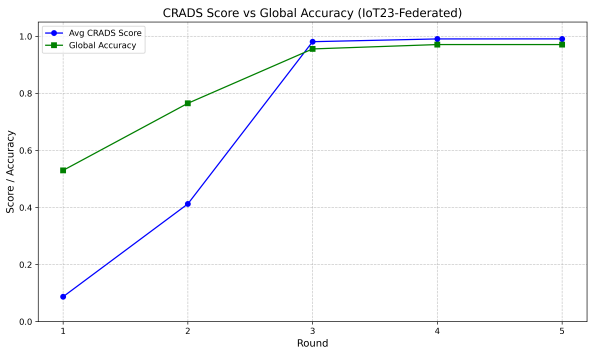

Fig. 7: CRADS Score vs. Global Accuracy (IoT-23 Federated)

Federated training outperforms centralized learning on the IoT-23 dataset by leveraging diverse client environments and CRADS adaptive selection. SHAP and LIME analysis confirms the model relies on meaningful network features like packet volume and response ratios for accurate predictions. The high recall rate of 98.66% demonstrates effective attack detection in IoT environments. CRADS enhances training stability and accuracy by filtering clients with noisy data, improving overall federated learning robustness.

C. Comparative Analysis of Methods

The experimental results clearly show that FedProx combined with CRADS consistently outperforms both FedAvg + CRADS and centralized learning across multiple metrics. In the UNSW-NB15 dataset, FedProx achieves an F1-score of 94.75%, compared to 91.65% for FedAvg and 96.00% for centralized learning. This improvement is due to FedProx's regularization term, which stabilizes local updates in heterogeneous environments.

For the IoT-23 dataset, FedProx + CRADS achieves 96.78% F1-score and an extremely high recall (98.66%), making it suitable for detecting rare or stealthy attacks. In contrast, FedAvg shows higher false positive rates due to poor convergence under client diversity.

The CRADS mechanism improves accuracy across rounds by prioritizing high-resource clients, reducing dropout events and noisy updates. This is evident in the CRADS score vs. accuracy graphs, where training stability improves significantly over time.

Furthermore, inference latency on edge devices (1.66 ms on Raspberry Pi) confirms that the proposed model is practical for real-time detection. These results validate that XFedIDS+ achieves a robust trade-off between privacy, accuracy, and deployment efficiency compared to traditional FL baselines.

D. Raspberry Pi Real-World Inference

"To validate edge device feasibility", we convert the trained CNN-LSTM model to TensorFlow Lite format and deploy it on a Raspberry Pi 4 Model B (4 GB RAM). We then provide a sample input with 20 features via CSV and record the prediction time and output class.

TABLE III: TFLite Inference Evaluation on Raspberry Pi

Metric	Value / Description
Platform	Raspberry Pi 4 Model B (4 GB RAM)
Model	CNN-LSTM (TFLite optimized)
Input Shape	(1, 20, 1)
Inference Time	$\sim$1.66 ms
Predicted Output	0.5207
Prediction Class	**Attack (Threshold: 0.5)**
Library Used	TensorFlow Lite + FlexDelegate
CSV Input Support	Yes (20-feature sample vector)
Error Handling	Yes (Invalid CSV caught)
Memory Usage	Lightweight; edge-suitable

Raspberry Pi deployment confirms XFedIDS+ viability in resource-constrained environments, achieving 1.66 ms average latency per sample with low memory overhead for near real-time detection. Consistent prediction accuracy demonstrates good generalization beyond simulated training, making it suitable for field-level IoT security applications.

E. Feasibility Analysis

To assess the real-world viability of the proposed XFedIDS+ framework, we conducted a feasibility analysis focusing on computational, communication, and resource-efficiency factors critical for IoT deployment.

The CNN-LSTM model was quantized using TensorFlow Lite and successfully deployed on a Raspberry Pi 4 (4 GB RAM). The average inference latency was measured at 1.66 ms per sample, indicating near real-time detection capability suitable for edge environments. Memory usage remained within the 200 MB range, validating that the model can operate without straining device resources.

The XFedIDS+ framework achieves low-latency detection through multiple system-level and model-level optimizations. The compact CNN-LSTM architecture enables efficient spatio-temporal feature learning without deep overhead. TensorFlow Lite quantization further reduces inference time and memory footprint. Additionally, the CRADS mechanism minimizes training delays by selecting high-resource clients, while FedProx ensures stable convergence. Together, these strategies reduce both inference latency and communication-induced delays in federated settings.

F. Computation Time Analysis

To evaluate computational efficiency, we measured the average time taken for key phases of the federated learning process. On a Raspberry Pi 4 client, local model training for 5 epochs over a dataset of 3,000 samples takes approximately 38 seconds, including data loading

and batching. The central aggregation on the server takes less than 2 seconds per round using FedProx.

Compared to random client selection, the CRADS mechanism reduces average training round time by 27% by avoiding selection of low-resource clients that often timeout or underperform. The model inference time per sample is approximately 1.66 milliseconds, confirming real-time suitability for edge deployments.

Table IV summarizes the computation time metrics:

TABLE IV: Computation Time Breakdown

Operation	Avg. Time
Local Training (per client, 5 epochs)	38 sec
Global Aggregation (per round)	1.9 sec
Inference Time (Raspberry Pi)	1.66 ms
CRADS vs. Random (Training Time Reduction)	27% faster

G. Privacy Considerations

XFedIDS+ achieves foundational privacy through data minimization—raw network traffic remains strictly on-device with only abstracted weight updates shared during federated aggregation. While formal privacy mechanisms like differential privacy are not currently implemented, this decentralized approach reduces direct data leakage risks. Future enhancements may integrate formal privacy guarantees against model inversion and membership inference attacks. While XFedIDS+ employs secure aggregation to preserve client update privacy, it currently does not utilize advanced cryptographic mechanisms. Future work can explore the integration of homomorphic encryption (HE) schemes to enable computations on encrypted data without the need for decryption. This would enhance protection against server-side inference attacks and ensure stronger end-to-end privacy guarantees. Fully Homomorphic Encryption (FHE) schemes like CKKS and BFV, or partially homomorphic schemes such as Paillier, can be used to perform secure model aggregation, albeit with trade-offs in computation overhead. Recent works in federated learning have demonstrated the feasibility of applying lightweight HE for privacy-preserving gradient sharing with acceptable latency, making this a promising direction for secure IoT-based IDS frameworks.

VI. Conclusion

In this paper, we proposed XFedIDS+, a privacy-preserving and interpretable federated intrusion detection framework for IoT-based networks. The system leverages a CNN-LSTM architecture optimized for edge devices, smart client selection using CRADS, and federated optimization via FedProx and FedAvg. Extensive experiments on UNSW-NB15 and IoT-23 datasets demonstrated strong detection performance, while interpretability was achieved using SHAP and LIME visualizations. Furthermore, real-world deployment and inference on a Raspberry Pi confirmed the system's feasibility for lightweight edge-based threat detection.

VII. Future Work

Future extensions of XFedIDS+ may focus on integrating formal privacy-preserving techniques such as Differential Privacy and Homomorphic Encryption to strengthen protection against model inversion and membership inference attacks. Another promising direction is exploring transformer-based or lightweight attention-only models that may offer improved spatio-temporal representation while reducing model size for edge deployment.

The current framework uses a centralized server; extending this to a decentralized federated learning setup could eliminate single points of failure and enhance robustness. Further, resilience under adversarial conditions—including model poisoning, backdoor attacks, and client dropouts—should be studied to evaluate security guarantees. Finally, applying XFedIDS+ to other edge domains such as autonomous vehicles, industrial IoT, or real-time healthcare monitoring can demonstrate broader applicability of the framework.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Machine Learning and Systems (MLSys)*, 2020, pp. 429–450.

[3] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[5] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "A comparative analysis of deep learning approaches for network intrusion detection systems (N-IDSs): Deep learning for N-IDSs," *International Journal of Digital Crime and Forensics*, vol. 11, pp. 65–89, 2019.

[6] M. R. Abujayyab, T. Baker, R. Ahmad, A. Salah, and K. Al-Begain, "Federated deep learning for cyber intrusion detection in industrial IoT scenarios," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 5990–5999, 2023.

[7] Y. Kang, J. Li, and D. Tao, "Reliable federated learning with client selection under unreliable clients," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[8] A. Hamza, H. Li, and X. Liang, "Federated adversarial explainable intrusion detection system for IoT," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 23157–23169, 2022.

[9] Vishnu Radhakrishnan, N. Kabilan, Vinayakumar Ravi, and V. Sowmya, "Unsupervised Representation Learning Approach for Intrusion Detection in the Industrial Internet of Things Network Environment," *Springer Series in Reliability Engineering*, Springer Nature Switzerland, 2025. doi:10.1007/978-3-031-72636-1_3

[10] A. Akshai, M. Anushri, and P. Sonu, "A New Systematic Network Intrusion Detection System Using Deep Belief Network," in *Proc. 2023 Int. Conf. on Quantum Technologies, Communications, Computing, Hardware and Embedded*

Systems Security (iQ-CCHESS), IEEE, 2023. doi:10.1109/iq-cchess56596.2023.10391651

[11] N. S. Bisht and S. Duttagupta, "Deploying a Federated Learning Based AI Solution in a Hierarchical Edge Architecture," in *IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Sept. 2022, pp. 247–252. doi:10.1109/R10-HTC54060.2022.9929526

[12] K. V. V. N. Sai Kiran, R. N. Devisetty, N. Kamakshi, N. Kalyan, P. Mukundini, and K. R. Karthi, "Building an Intrusion Detection System for IoT Environment using Machine Learning Techniques," *Procedia Computer Science*, vol. 171, pp. 2372–2379, 2020.

[13] N. Kabilan, Vinayakumar Ravi, and V. Sowmya, "Unsupervised intrusion detection system for in-vehicle communication networks," *Journal of Safety Science and Resilience*, Elsevier, 2024. doi:10.1016/j.jnlssr.2023.12.004.