IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 September 2025, accepted 18 September 2025,
date of publication 23 September 2025, date of current version 10 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3613459

RESEARCH ARTICLE

Federated Learning With Explainable AI for Malicious Traffic Detection in IoT Networks

MUHAMMAD AHMAD BILAL[1], IHTESHAM UL ISLAM[1], NAIMA ILTAF[1], MUHAMMAD JUNAID KHAN[2], AND M. JALEED KHAN[3]

[1]Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad 46000, Pakistan
[2]Department of Electrical Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad 46000, Pakistan
[3]Data Science Institute, University of Galway, Galway, H91 TK33 Ireland

Corresponding author: M. Jaleed Khan (m.khan12@universityofgalway.ie)

ABSTRACT The rapid proliferation of Internet of Things (IoT) devices has intensified the need for scalable, privacy-preserving, and interpretable intrusion detection systems (IDS). In this work, we propose a novel framework that combines Federated Learning (FL) with Explainable Artificial Intelligence (XAI) to detect malicious traffic across distributed IoT networks without centralizing sensitive data. Our IDS model employs a lightweight deep neural network trained collaboratively using FedAvg. SHAP provides global and local attributions; LIME supplies instance-level attributions complementary to SHAP. Experiments conducted on three benchmark datasets: Edge-IIoT, CIC-IoT2023, and TII-SSRC-23, demonstrated that our approach achieves 99.3%, 99.5% and 99.0% accuracy in binary classification, respectively and 97.2%, 98.0% and 96.5% accuracy in multi-class scenarios, respectively, closely matching centralized models while preserving data locality and data privacy. Moreover, the integrated XAI methods enhance the model's transparency by identifying key traffic features contributing to each alert. These results establish FL-XAI as a practical, interpretable, and privacy-respecting IDS for real-world IoT deployments.

INDEX TERMS Intrusion detection (ID), FL, XAI, deep learning (DL), DoS, DDoS.

I. INTRODUCTION

The IoT continues to expand rapidly, with estimates of more than 75 billion IoT devices in use by 2025. This explosive growth of connected sensors, appliances, and industrial controllers is accompanied by a ''trove of security concerns'', as poorly secured IoT devices become targets and vectors for cyberattacks [1]. High-profile incidents such as the Mirai botnet – which compromised hundreds of thousands of IoT devices to launch massive DDoS attacks – have demonstrated the disruptive potential of IoT-based threats [2]. In the Mirai case, infected cameras and routers were co-opted into a botnet that overwhelmed domain name services, highlighting how malicious IoT traffic can have far-reaching impacts on Internet infrastructure. These trends underscore the urgent need for effective IDS tailored to IoT networks.

Machine learning (ML) has emerged as a core enabling technology for modern network ID, given its ability to recognize patterns of known attacks and anomalies in large traffic datasets [3]. A number of recent works have applied supervised and DL techniques to IoT security with promising results. For example, Zeeshan et al. [4] achieved 96%+ accuracy using a deep neural network to detect DoS/DDoS attacks in merged IoT traffic datasets (UNSW-NB15 and Bot-IoT). Ahmad et al. [5] similarly attained high detection rates (over 97% multi-class accuracy) by training ML models (Random Forest, SVM, ANN) on curated IoT traffic features from the UNSW-NB15 dataset. These studies demonstrate the efficacy of data-driven ID in IoT environments. However, a common assumption in such approaches is that training data from distributed IoT devices can be centralized in a cloud or data center for model learning. In practice, aggregating raw IoT traffic data at a central server raises serious privacy, bandwidth, and scalability concerns [3]. Sensitive information from cameras or medical sensors,

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

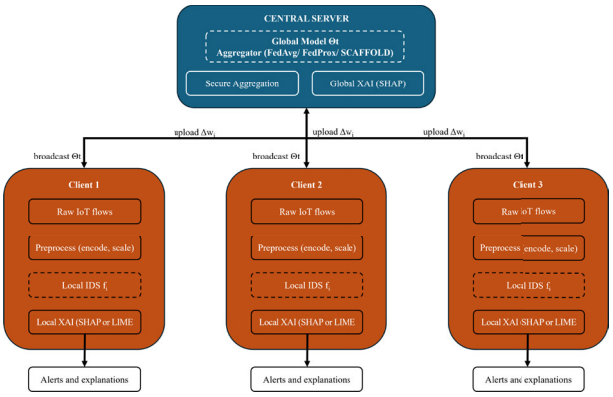

FIGURE 1. System overview illustrating FL with local client models and a central aggregator, including the flow of XAI insights between clients and the global model.

for instance, may be subject to privacy regulations, and resource-constrained IoT networks often cannot afford the bandwidth to continuously ship high-volume traffic logs to the cloud.

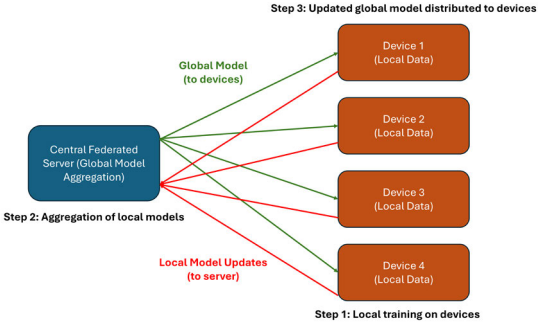

FIGURE 2. FL Communication Flow.

To address these challenges, FL has emerged as a promising paradigm for distributed ML in IoT settings [6]. The Fig. 1 clearly depicts the interaction between federated clients and the central server, and also highlights how explainability (XAI) is integrated. In FL, a shared global model is trained collaboratively by aggregating model updates from multiple devices that each train locally on their own data, rather than by collecting the data centrally (Fig. 2). This approach preserves data privacy and reduces communication costs, since only model parameters or gradients (not raw data) are exchanged [7]. FL has been successfully applied to IoT security tasks in recent studies. For example, DIoT by Nguyen et al. is a federated self-learning anomaly detection system for IoT that builds device-specific communication profiles and detects deviations without centralized training data [8]. DIoT achieved a 95.6% detection rate for compromised IoT devices (e.g. Mirai-infected devices) with no reported false alarms in a real smart home deployment. In the industrial domain, Li et al. proposed DeepFed, a federated DL framework for ID in Industrial Control Systems, which showed that a global model can be learned from distributed industrial sites with minimal loss in accuracy compared to central training [9], [10]. Likewise, a comparative study by Rahman et al. concluded that federated and on-device learning approaches can substantially enhance IoT ID while maintaining data privacy, compared to traditional centralized IDS [6]. These works indicate that FL is a viable approach to enable scalable, privacy-preserving IDS in large IoT networks.

Despite progress in detection accuracy and privacy-preserving training, a critical gap remains: lack of explainability in machine-learned intrusion detectors. Advanced DL models are often treated as "black boxes" that output an alert without human-understandable justification [11]. In security applications, this opaqueness can hinder trust and adoption – network administrators are less likely to deploy an IDS whose decisions cannot be interpreted or validated. The inability to explain why a particular network flow was classified as malicious also makes it difficult to debug false positives or to glean actionable intelligence about new attack patterns. Recognizing this problem, the research community has turned to XAI techniques to bring transparency to IDS models. XAI methods seek to provide human-interpretable explanations (e.g. feature importance scores, decision rules) for individual predictions or overall model behavior in complex AI systems. In the context of ID, recent studies have begun integrating XAI to illuminate how ML models identify threats. For instance, Mohale and Obagbuwa [11] present a systematic review on XAI in IDS, highlighting that rule-based and feature-importance methods can help analysts understand and trust IDS decisions. Two popular model-agnostic XAI approaches are LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations). LIME builds simple surrogate models (e.g. linear models or decision trees) locally around a specific prediction to explain which features drive the model's decision [12]. SHAP, on the other hand, uses principles from cooperative game theory to quantify each feature's contribution to a given prediction, offering consistent and theoretically grounded attributions [13]. Both methods have been applied to interpret ID models. For example, Gaspar et al. [14] used LIME and SHAP to explain an IoT malware detection model's outputs, helping to identify the network features most indicative of attacks. Similarly, Corea et al. [3] introduced an explainable DL framework for IoT networks that employed SHAP to reveal the inner workings of their IDS model, improving clarity for security analysts. These efforts show that XAI can enhance transparency in standalone IDS. However, the integration of XAI with FL for IoT ID remains largely unexplored. A very recent study by Kalakoti et al. [15] has begun to address this, proposing an explainable FL approach for IoT botnet detection using SHAP to interpret the global model. This reinforces the timeliness and novelty of combining FL and XAI for IoT security, as pursued in our work.

In this paper, we present a unified framework that brings together FL and XAI for malicious traffic detection in IoT networks. To our knowledge, this is one of the first comprehensive studies to apply XAI techniques to a federated

IoT IDS. The key contributions of our work are summarized as follows:

- **Federated IoT ID Architecture:** We design a system architecture where distributed IoT edge devices collaboratively train a deep ID model under a FL scheme. The architecture addresses IoT challenges of data privacy, limited bandwidth, and device heterogeneity by keeping raw data on-device and sharing only model updates. We detail the novel integration of an XAI module into this federated IDS to provide real-time explanations for model outputs.
- **XAI Integration:** We incorporate model-agnostic explainability methods (specifically SHAP and LIME) to interpret the federated model's predictions. Our framework generates per-instance explanations of why a network flow is classified as attack or benign, and highlights which traffic features are most influential. We demonstrate that these explanations align with domain intuition (e.g., certain packet rates or protocol flags indicating attacks), thereby enhancing the transparency and trustworthiness of the IDS. To the best of our knowledge, our work is the first to apply SHAP and LIME in the context of federated ID for IoT.
- **Evaluation on Diverse IoT Attack Datasets:** We conduct extensive experiments on three recent, comprehensive IoT/IIoT ID datasets – Edge-IIoT [16], CIC-IoT2023 [17], and TII-SSRC-23 [2] – which together cover a wide spectrum of attack types (DDoS, DoS, Man-in-the-Middle, scans, malware, etc.), device types, and network protocols. By evaluating on multiple benchmarks, we demonstrate the generality of our FL-XAI approach. Our results show high detection performance (e.g., 97–99% accuracy, recall, and F1-score for attack detection) across all datasets, indicating robustness against diverse IoT threats.
- **Comprehensive Results and Analysis:** We present detailed experimental results including confusion matrices, performance metrics, and visualization of feature importance derived from XAI. We compare the federated model's performance against a centralized training baseline and observe negligible degradation (<1% difference in F1-score), validating the efficacy of FL. The explainability analysis provides insight into the model's decision process; for instance, we identify which features (such as traffic volume, port numbers, or OPC-UA function codes in IIoT scenarios) are most associated with specific attacks in each dataset. We also discuss the computational overhead of XAI and its trade-offs.
- **Comparison with State-of-the-Art:** We benchmark our approach against existing ID works. In particular, we show that our federated model's accuracy is on par with or better than several centralized IDS methods from recent literature, while offering the advantage of data privacy. We also outperform prior federated IDS like DIoT in terms of detection coverage (multi-attack classification vs. binary anomaly detection) and provide explanatory capabilities which those methods lack. A qualitative comparison with related work is provided to highlight the novelty of combining FL and XAI in our solution.

The remainder of the paper is organized as follows: Section II presents the related work in federated and explainable IDS systems. Section III provides an overview of the proposed architecture. Section IV details the datasets, preprocessing, and model training configuration. Section V outlines the experimental setup and evaluation results, including explainability analysis and a comparison with existing approaches. Finally, Section VI concludes the paper and discusses potential directions for future research.

II. RELATED LITERATURE

A. IoT ID AND DATASETS

IoT ID research has evolved significantly, transitioning from legacy datasets like DARPA'98/KDD'99 [18], [19], which lacked IoT specificity, to modern datasets such as UNSW-NB15 [20] and Bot-IoT [21]. Ahmad et al. [5] demonstrated high accuracy (98.67%) using Random Forests on UNSW-NB15, and Zeeshan et al. [4] combined UNSW-NB15 and Bot-IoT datasets using bi-directional LSTMs, achieving over 96% accuracy. Recently, comprehensive datasets like Edge-IIoT [16], CIC-IoT2023 [17], and TII-SSRC-23 [2] have emerged, offering realistic IoT attack scenarios and varied device traffic, significantly enriching IDS benchmarks.

Unlike centralized approaches, our study leverages FL on these modern datasets to address privacy and incorporates XAI, providing clear rationales for IDS alerts.

B. FEDERATED AND DISTRIBUTED LEARNING FOR IDS

FL, pioneered by FedAvg (McMahan et al. [22]), enables decentralized model training without raw data sharing, ideal for IoT contexts due to privacy and bandwidth considerations. Early implementations like DIoT (Nguyen et al. [8]) utilized local autoencoders aggregated centrally to detect IoT anomalies effectively. DeepFed (Li et al. [23]) demonstrated high accuracy for intrusion classification in Industrial IoT. Advanced FL architectures, including hierarchical and clustered approaches [24], further improve accuracy. Surveys (Abdulrahman et al. [25]) suggest FL's balance between centralized and on-device methods offers optimal trade-offs. Our research extends this literature, using advanced datasets and incorporating XAI, addressing data distribution heterogeneity, resource constraints, and class imbalance.

C. XAI IN ID

XAI methods are critical for establishing trust in IDS, with popular approaches including feature attribution techniques like SHAP and LIME, as well as inherently interpretable models such as decision trees. For example, Keshk et al. [26] developed an explainable DL–based IDS for IoT that

integrates SHAP and related interpretability tools to clarify model decisions, achieving high accuracy across NSL-KDD, UNSW-NB15, and TON-IoT datasets. Similarly, Gaspar et al. [14] conducted a comparative study of SHAP and LIME on multiple machine-learning IDS models, demonstrating that SHAP effectively highlights critical attack-indicative features. Building on this, Gyawali et al. [27] proposed an XAI-enhanced IoT IDS that employs SHAP to generate both instance-level and global explanations, thereby improving analysts' situational awareness. In our federated IDS framework, we likewise deploy SHAP and LIME to generate interpretable explanations at the server—where computational resources suffice—supporting analysts' decisions without imposing undue overhead on resource-constrained edge devices.

D. FL FOR DISTRIBUTED IoT SECURITY

FL collaboratively trains global models without sharing local IoT data, beneficial for privacy-sensitive environments. The foundational FedAvg algorithm (McMahan et al.) iteratively averages local updates from clients [22]. IoT-specific challenges addressed by FL include non-IID data distributions, resource constraints, communication limitations, and privacy risks. Techniques such as adaptive aggregation and hierarchical FL help mitigate these issues [28]. Our implementation uses lightweight models suitable for resource-constrained edge nodes, testing robustness under varied data distributions and exploring distributed attack pattern detection.

E. XAI FOR NETWORK TRAFFIC CLASSIFICATION

XAI enhances IDS by clarifying ML-driven decisions. Local interpretability via LIME (Ribeiro et al. [29]) explains individual predictions through surrogate models. SHAP (Lundberg et al. [13]) consistently quantifies feature impacts. Global explanations aggregate local insights, revealing influential features and patterns. Integrating XAI into federated IDS addresses trust, debugging, compliance, and sensor relevance [30]. By offering transparent alerts, XAI helps administrators validate and act confidently, guiding model refinement and enhancing forensic investigations. Our approach strategically combines FL with SHAP and LIME, optimizing interpretability and computational feasibility in IoT contexts.

F. GAPS AND OUR CONTRIBUTIONS

To the best of our knowledge, there is limited work that jointly addresses (i) federated training on modern, heterogeneous IoT datasets, and (ii) integration of model-agnostic XAI techniques for transparent, per-instance threat interpretation. Existing FL-based IDS frameworks often sacrifice interpretability, while XAI-driven IDS research relies on centralized architectures and outdated datasets.

Our work fills this critical gap by introducing a fully decentralized IDS model that combines FL with SHAP and LIME to support real-time explainability. We evaluate our system across three diverse and recent IoT datasets—Edge-IIoT, CIC-IoT2023, and TII-SSRC-23—under both IID and non-IID scenarios, and demonstrate high classification performance with detailed explainability outputs. This integration advances the current state-of-the-art in secure, transparent, and deployable ID for IoT environments.

III. PROPOSED METHODOLOGY

In this section, we detail the methodology for our FL-based IDS and how XAI techniques are employed. We begin by describing the FL system architecture, preprocessing, then the intrusion detection model and features. We then outline the federated training algorithm (with pseudocode) that trains the model across distributed clients. Finally, we describe how we generate and interpret explanations (SHAP and LIME) from the trained model's predictions.

A. SYSTEM ARCHITECTURE

Our FL architecture adopts a star topology integrated with an explainability module at the central aggregator (Fig. 3). The architecture comprises:

- **IoT Devices / Edge Nodes**: Serve as local clients collecting data and running local models. Clients simulate sensors or edge nodes aggregating sensor data. We simulate these by partitioning datasets for each virtual client.
- **Central Aggregator (FL Server)**: Orchestrates FL rounds by initializing, distributing, aggregating models, and computing explanations centrally. It is assumed to have sufficient computational resources.
- **Federated Communication Network**: Securely transfers model parameters between clients and the aggregator. While we simulate ideal communication, actual deployments require encryption and authentication for security.
- **Security hardening**: The aggregator accepts only authenticated updates over TLS, performs robust anomaly checks on incoming updates using cosine-simi larity and coordinate-wise median filters, and rejects outliers before aggregation. We enable secure aggregation to hide individual client contributions. Optional differential privacy with per-client clipping provides formal privacy guarantees against model inversion attacks. These mechanisms mitigate poisoning and inference risks without changing the core FL-XAI workflow.
- **Explainability Module**: Operates centrally, generating explanations (SHAP values, LIME) from global model predictions. Explanations may be computed centrally or locally on clients for efficiency and privacy. We primarily use central explanations.
- **Security Analyst Interface**: Displays alerts and explanations clearly to security analysts. Though not fully implemented, we simulate this by narrative explanations derived from SHAP/LIME analyses.

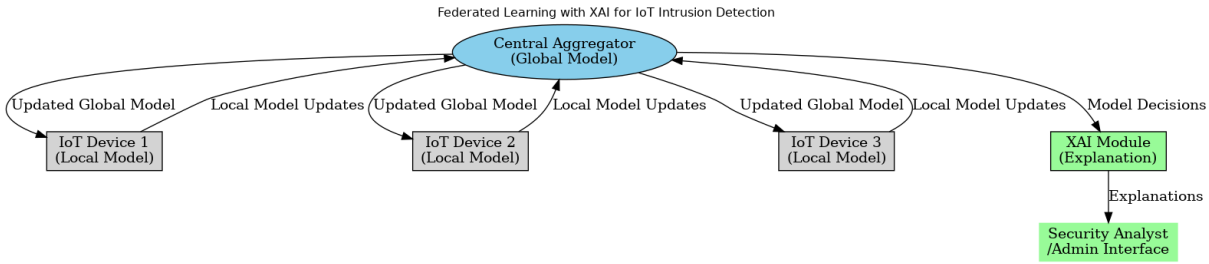

FIGURE 3. High-level architecture of the proposed FL-XAI intrusion detection system, showing federated client training with a central aggregator and integrated XAI module.

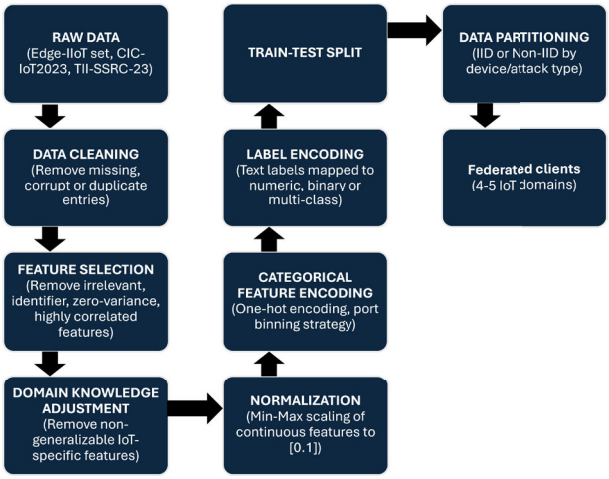

FIGURE 4. Flow of data preprocessing and feature representation for IoT IDS.

The system operates in two phases:

- **Training Phase**: The server initializes a global model, sends it to clients for local training, aggregates updates (FedAvg), and repeats until convergence. Validation using XAI helps ensure the model focuses on relevant features.
- **Detection Phase**: The trained global model is deployed for live ID. Alerts trigger the XAI module to generate explanations either locally or at the aggregator, based on resource availability and privacy considerations.

Our architecture is flexible, supporting various models (e.g., neural networks, Random Forests) and explainability methods (SHAP, LIME), enabling adaptability to future advancements. Deployment could leverage existing FL and XAI libraries like TensorFlow Federated, PySyft, and SHAP.

B. DATA PREPROCESSING AND FEATURE REPRESENTATION

Prior to training, all three datasets (Edge-IIoT, CIC-IoT2023, TII-SSRC-23) underwent standardized preprocessing to ensure consistency and model readiness (see Fig. 4).

- **Data Cleaning:** First, missing or corrupt records were removed. For Edge-IIoT, less than 1% of the data contained anomalies such as undefined service names or negative packet counts, which were discarded. CIC-IoT2023 had fewer missing entries, estimated at below 0.5%, while TII-SSRC-23 exhibited a slightly higher rate of approximately 0.7%, including some duplicated entries which were filtered out to prevent data leakage or skewed learning.
- **Feature Selection:** Irrelevant identifiers (e.g., flow ID, timestamps) and constant-valued features were dropped. Highly correlated pairs (correlation >0.95) were pruned to minimize multicollinearity. Dataset-specific but non-generalizable attributes (e.g., IoT device type) were excluded to prevent model bias. Final feature counts were: 40 (Edge-IIoT), 45 (CIC-IoT2023), and 52 (TII-SSRC-23).
- **Normalization and Encoding:** Continuous features were scaled to [0,1] using min-max normalization. Mathematically normalization is given as per eq 1.

$$x_{scaled} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x is the original feature value and x_{scaled} is the normalized value. Categorical fields like protocol type were one-hot encoded. Port numbers were binned into ''well-known,'' ''registered,'' and ''dynamic'' categories to avoid sparsity in encoding.
- **Class Imbalance:** We use a cost-sensitive objective at each client. Let p_c be the class prior on the client. We set class weights $w_c = 1/\log(1 + \alpha p_c)$ with $\alpha = 10$ to damp the effect of extreme majorities, and optimize a weighted cross-entropy. For very skewed families we also evaluated focal loss with focusing parameter $\gamma = 2$ and balance parameter β. In CIC-IoT2023 we limit Synthetic Minority Over-sampling Technique (SMOTE) [31] to the tiny Web and BruteForce classes to avoid synthetically inflating already dominant families. This hybrid strategy reduces bias while avoiding over-optimistic gains from heavy oversampling.
- **Label Encoding:** All textual class labels were mapped to integers. For binary classification: benign = 0, attack = 1. For multi-class classification:
  - CIC-IoT2023 used 8 labels (7 attack types + benign),

- Edge-IIoT used 6 labels (5 attack categories + benign),
- TII-SSRC-23 used 5 aggregated categories (including benign) for practical class support, as fine-grained classification yielded data imbalance.

Post-preprocessing, datasets were split into training and test sets. To simulate real-world distributed learning, data was partitioned across 4–5 federated clients.

- **IID partitioning:** Data was randomly and evenly distributed across clients, ensuring similar class distributions.
- **Non-IID partitioning:** Data was grouped by device type or attack category. For TII-SSRC-23, traffic from similar device groups was clustered on specific clients to mimic realistic deployment conditions.

This setup allows us to evaluate the federated model's robustness under both balanced and skewed local data distributions.

C. ID MODEL

Given the processed feature vectors as input, we needed to choose a ML model architecture that is expressive enough to capture complex attack patterns, yet efficient enough for federated training on edge nodes. We opted for a deep neural network (DNN) classifier – specifically, a multi-layer perceptron – for our IDS model. A neural network provides non-linear decision boundaries capable of capturing subtle differences between malicious and benign traffic, which simpler models might miss [3]. Additionally, neural networks can be readily partitioned and aggregated in FL (since we just average weights), and they are amenable to gradient-based explainability methods.

The architecture of our neural network model is as follows:

- Input layer: size = n features (after preprocessing, n≈40-52 as mentioned). Each feature of a network flow enters as a normalized value.
- Hidden Layer 1: a fully-connected dense layer with 128 neurons, ReLU activation. We include a dropout of 20% after this layer to improve generalization (especially needed since some datasets like CIC-IoT2023 have millions of samples which could lead to overfitting without regularization).
- Hidden Layer 2: a fully-connected layer with 64 neurons, ReLU activation. (We found that two hidden layers were sufficient; adding more did not yield significant improvement but did increase communication cost. This 2-layer MLP is also lightweight for edge devices.) We again use dropout (20%) here during training.
- Output Layer: a fully-connected layer with C neurons, where C is the number of classes (for binary classification, C=1 with a sigmoid activation; for multi-class, C equals number of attack categories + 1 for benign, with a softmax activation). This layer produces the probability (or logit scores) for each class.

We initialize weights using Xavier/Glorot initialization. The model has on the order of a few thousand weight parameters. Model features are shown in table 1.

Training Hyperparameters: We train the model using the Adam optimizer (an adaptive learning rate gradient descent) with an initial learning rate of 0.001. A cross-entropy loss function is used (binary cross-entropy for binary case, categorical cross-entropy for multi-class). Each client in federated training runs a fixed number of local epochs per round (we typically set local epochs E = 1 or 2, meaning each client goes through its data once per round, which is common in FL to balance local computation vs. communication) [22]. In FL the effective number of parameter updates equals $E \times T$, where E is local epochs per round and T is the number of rounds. We set $E \in 1, 2$ to reduce client drift and communication cost, and run for tens of rounds until the validation loss on the server plateaued with early stopping (patience 5 rounds). Learning curves for all three datasets show monotone improvement and no overfitting under this schedule, which empirically validates that one or two local passes per round are sufficient for the reported results. The batch size for local training is 64. We found that a smaller batch (like 32) sometimes helped converge slightly faster, but 64 was a good compromise and is also more efficient on vectorized hardware.

During training, we monitor performance on a validation split of data. In centralized training (for baseline comparison), we would use something like 80% train, 10% validation, 10% test. In federated training, since data is distributed, each client can keep a portion of its local data as local validation, and/or the server can have a small global validation set (perhaps some public IoT attack data or an out-of-band dataset) to evaluate the global model after each round. In our experiments, for simplicity, we set aside 10% of the overall training data as a validation set at the server (this is not seen by clients in training, just used for monitoring global model performance between rounds). This validation set is also used for computing global SHAP explanations at the end.

D. FEDERATED TRAINING ALGORITHM

We now formalize the federated training process in pseudo-code, which closely follows the standard FedAvg algorithm with a few additions for explainability monitoring (algorithm 1).

FedAvg mathematically is given by eq 2. Where W^t is the global model at round t, $W^{t,k}_{local}$ is the locally trained model at client k and K is the total number of federated clients.

$$W^t = W^{t-1} + \frac{1}{K}\sum_{k=1}^{K}(W^{t,k}_{local} - W^{t-1}) \tag{2}$$

FL is vulnerable to gradient inversion, poisoning, and Byzantine behaviors. When simulating deployments we apply per-client gradient clipping and norm bounding before aggregation. The server aggregates updates using a secure-aggregation primitive that hides individual client

TABLE 1. Architecture of the DNN model used for the IDS.

Dataset	Input Layer (Neurons)	Hidden Layer 1 (Neurons)	Hidden Layer 1 (Parameters)	Hidden Layer 2 (Neurons)	Hidden Layer 2 (Parameters)	Output Layer (Neurons)	Output Layer (Parameters)	Total Parameters	Activation Functions
Edge-IIoT	40	128	5248	64	8256	6	390	13894	ReLU (hidden), Softmax (output)
CIC-IoT2023	45	128	5888	64	8256	8	520	14664	ReLU (hidden), Softmax (output)
TII-SSRC-23	52	128	6784	64	8256	5	325	15365	ReLU (hidden), Softmax (output)

Algorithm 1 FL for IoT IDS with XAI Monitoring

Input: Global model initial parameters W^0; Clients $1, \ldots, K$ with local datasets $D_1, \ldots, D_K$; Learning rate η; Local epochs E; Rounds T
Output: Trained global model W^T
for *each round* $t = 1$ *to* T **do**
  Server broadcasts current model $W^{(t-1)}$ to all clients;
  for *each client* $k = 1$ *to* K ***in parallel*** **do**
    Load $W^{(t-1)}$ into local model;
    for *epoch* $= 1$ *to* E **do**
      for *each batch* b *in* D_k **do**
        Compute gradients $g = \nabla_W \text{Loss}(W, b)$
        Update local weights:
        $W_{\text{local}} = W_{\text{local}} - \eta \cdot g$;
    Compute local update: $\Delta W_k = W_{\text{local}} - W^{(t-1)}$;
  Server receives $\Delta W_1, \ldots, \Delta W_K$ from clients;
  `// Secure Aggregation (simplified)`
  $W^t = W^{(t-1)} + \frac{1}{K} \sum_{k=1}^{K} \Delta W_k$
  `// If using data-size weighted aggregation:`
  $W^t = W^{(t-1)} + \sum \frac{n_k}{\sum n_j} \cdot \Delta W_k$
  Server evaluates W^t on validation set;
  if *XAI monitoring is enabled* **then**
    for *sample of validation instances* **do**
      Compute SHAP values (or other XAI) using W^t;
      Check for anomalies in feature attributions;
  if *convergence criteria met* **then**
    break;
return (**final model** W^T);

updates and reveals only the sum. We chose the standard FedAvg aggregator for its simplicity and proven efficacy in federated settings. Despite the known challenges with non-IID data, FedAvg yielded excellent performance in our experiments (less than 1% accuracy gap to centralized, as shown in table 12). We now explicitly note that more robust aggregation strategies (e.g., FedProx, which adds a proximal term to handle heterogeneous data, or other robust aggregators) could further improve resilience to non-IID distributions or malicious clients [32]. However, in our setup FedAvg was sufficient to achieve near-centralized performance, as demonstrated in section V-E. In our implementation, we enabled XAI monitoring at certain milestones (for example, after training completes, and occasionally at intermediate rounds to ensure training is on track). This is not a standard part of federated algorithms, but we found it useful for research to inspect the model's behavior. In a real deployment, one might not do SHAP during training due to overhead, but could do a final SHAP analysis on the trained model.

E. EXPLAINABILITY INTEGRATION

To enhance trust and operational transparency, we integrate two model-agnostic explainability methods—SHAP and LIME—after federated model convergence. These tools are applied to interpret model decisions during inference:

- **SHAP (KernelSHAP)**: Computes per-feature Shapley values by comparing model output with a benign-only background distribution (100 samples). For a feature set $S \subseteq F$:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \tag{3}$$

  where ϕ_i is the SHAP value for feature i, $f_S(x_S)$ is the model prediction using features S and F is the total number of features. This highlights deviations from normal traffic and supports both instance-level and global explanations.
- **LIME**: Provides local surrogate models using perturbed inputs to approximate the decision boundary near a specific prediction. LIME explanations for an instance x are derived as follows:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{4}$$

  where f is the original complex model, g is an interpretable local surrogate model, π_x is the local neighborhood around x, L is a loss function measuring how close g approximates f and $\Omega(g)$ denotes model complexity regularization.

F. MODEL DEPLOYMENT AND DETECTION

After federated training, the final global model is deployed on IoT edge nodes for inference. When new network flows are observed, each node extracts relevant features and classifies the flow as benign or malicious, triggering alerts if malicious activity is detected. For each alert, explainability is integrated using SHAP and LIME, providing local explanations highlighting the critical features influencing the

TABLE 2. Class distribution of the Edge-IIoT dataset samples used in our experiments (80/20 split by class).

Class	Samples	%	Train	Test
Benign (Normal)	11,223,940	53.6	8,979,152	2,244,788
DoS/DDoS	8,365,122	39.92	6,692,098	1,673,024
Info-Gathering	1,222,819	5.84	978,255	244,564
Injection	67,118	0.32	53,694	13,424
Man-in-the-Middle	1,229	0.01	983	246
Malware	73,421	0.35	58,737	14,684
Total	**20,953,649**	**100**	**16,763,219**	**4,190,730**

model's decision. SHAP produces feature-level attributions indicating their contribution to the prediction, while LIME provides simplified linear explanations. Agreement between SHAP and LIME increases confidence in the generated explanation, whereas disagreement may suggest a need for deeper analysis.

Each alert generated is presented with a clear, actionable explanation to assist security analysts. For instance, an alert from the CIC-IoT2023 dataset might specify: "Predicted as DDoS attack with 99.2% confidence. Explanation: high packet rate (SHAP +0.75), large outbound byte volume (SHAP +0.10), and absence of DNS queries (SHAP -0.05)." This explanation clarifies the factors influencing classification, such as abnormal traffic volume and missing typical benign traffic features. Importantly, these explanations serve a post-hoc interpretive purpose without influencing the model's classification decisions directly, ensuring that detection accuracy remains uncompromised while significantly enhancing analyst understanding and trust.

IV. EXPERIMENTAL SETUP AND DATASETS

We evaluate our approach on three public IoT/IIoT ID datasets: Edge-IIoT, CIC-IoT2023, and TII-SSRC-23, as introduced earlier. Here we provide additional details on how we used these datasets in our experiments, the FL setup (number of clients, data partitioning, etc.), and the computing infrastructure used for training.

A. DATASETS

1) EDGE-IIoT (2022)

The Edge-IIoT [16] dataset includes approximately 20 million network flows captured from a variety of consumer and industrial IoT devices under 14 attack scenarios grouped into five broader categories: DoS/DDoS, Information Gathering, MITM, Injection Attacks, and Malware (table 2). We used 40 preprocessed features per flow, focusing on statistical metrics and protocol-level indicators. In our federated setup, data was partitioned among five clients by device type to simulate non-IID scenarios, with each client exposed to specific attack distributions. This setup reflects realistic deployment environments where different IoT domains encounter distinct threat patterns.

2) CIC-IoT2023 (2023)

The CIC-IoT2023 [17] dataset consists of 46 million flows spanning 33 attack types organized into seven families such as DDoS, DoS, Reconnaissance, Mirai, and Spoofing (table 3). It was collected using 105 IoT devices performing attack and victim roles, making it a highly realistic large-scale dataset. We used approximately 45 flow-based features, including protocol-level identifiers and traffic statistics. Data was distributed across four federated clients biased towards specific attack families, creating a challenging non-IID setup.

TABLE 3. Class distribution of the CIC-IoT2023 dataset samples used in our experiments (80/20 split by class).

Class	Samples	%	Train	Test
DDoS	33,984,560	72.79	27,187,648	6,796,912
DoS	8,090,738	17.33	6,472,590	1,618,148
Mirai	2,634,124	5.64	2,107,299	526,825
Benign	1,098,195	2.35	878,556	219,639
Spoofing	486,504	1.04	389,203	97,301
Recon	354,565	0.76	283,652	70,913
Web	24,829	0.05	19,863	4,966
BruteForce	13,064	0.03	10,451	2,613
Total	**46,686,579**	**100.00**	**37,349,262**	**9,337,317**

TABLE 4. Class distribution of the TII-SSRC-23 dataset samples used in our experiments (80/20 split by class).

Class	Samples	%	Train	Test
Benign	1,301	0.02	1,041	260
Brute force	35,172	0.41	28,138	7,034
DoS	7,490,929	86.53	5,992,743	1,498,186
Info-Gathering	1,038,363	11.99	830,690	207,673
Mirai	91,002	1.05	72,802	18,200
Total	**8,656,767**	100.00	**6,925,414**	**1,731,353**

3) TII-SSRC-23 (2023)

The TII-SSRC-23 [2] dataset, with 8.6 million samples across 5 broader types (table 4), covers five device categories (e.g., ICS, smart cameras) and includes specialized protocol features. We selected 52 features and assigned one device type per client in a strongly non-IID setting. This setup allowed us to test the federated model's generalization across heterogeneous traffic and attack vectors, highlighting the framework's adaptability and the utility of XAI in multi-domain contexts.

For all three datasets, we applied an 80/20 train-test split, with the training data further partitioned among federated clients to simulate realistic non-IID scenarios. In Edge-IIoT, five clients were assigned data based on device types, each observing a limited subset of attacks, reflecting domain-specific exposure. CIC-IoT2023 was divided among four clients by attack family, with each client biased toward specific threats like DDoS or Mirai, challenging the model to generalize across skewed distributions. TII-SSRC-23 was partitioned by device category, assigning traffic from smart cameras, ICS, and other devices to separate clients—representing a strongly non-IID setup. This allowed us to rigorously evaluate the robustness of our FL model and its explainability components across diverse IoT threat landscapes.

TABLE 5. Dataset characteristics. Entries marked "–" indicate information (e.g. total flows) not directly provided in source papers, but each dataset is large-scale as described.

Dataset	Year	IoT Scenario	Devices / Traffic Types	Attack Categories	Samples	Features
Edge-IIoT [16]	2022	Heterogeneous IoT/IIoT	10+ devices	14 (DoS, etc.)	20M	40
CIC-IoT2023 [17]	2023	Smart-home lab (105 devices)	105 devices (real)	33 (DDoS, etc.)	46M	45
TII-SSRC-23 [2]	2023	Mixed network traffic	8 types (audio, etc.)	26 (Bruteforce, etc.)	8.6M	52

TABLE 6. Approximate communication cost with 32-bit weights.

Dataset	Params	Model (KB)	Per round/client (KB)	Total 50r, 5c (MB)
Edge-IIoT	13,894	54.27	108.55	26.50
CIC-IoT2023	14,664	57.28	114.56	27.97
TII-SSRC-23	15,365	60.02	120.04	29.31

Table 2, 3, 4 & 5 summarize key properties of these datasets. (We split each dataset among simulated clients as described below.)

B. EXPERIMENTAL PLATFORM

All experiments, including model training and inference with explainability, were conducted using a combination of local hardware and cloud resources. The central server was configured with an Intel Core i7 CPU (8 cores) and 16 GB RAM, simulating multiple federated clients sequentially on the same machine to maintain algorithmic fidelity. For larger workloads such as training on the CIC-IoT2023 dataset, Google Colaboratory's GPU (Tesla K80 with 12 GB RAM) was used to accelerate training, reducing per-round time by approximately 4×. SHAP and LIME computations were performed on CPU, with SHAP explanations taking roughly 0.2 seconds per instance and LIME taking around 0.5 seconds. While not optimized for real-time deployment, this performance is adequate for post-alert interpretation in IoT environments.

The implementation was done entirely in Python using TensorFlow 2.x/Keras for model development and a manually implemented FedAvg loop for FL. Explainability was integrated using the SHAP library (v0.41) and the lime package. Data preprocessing was managed using NumPy, Pandas, and scikit-learn. This custom setup enabled precise control over the federated training and interpretability pipeline, ensuring that insights remained transparent and replicable throughout the experimental workflow.

1) COMMUNICATION AND RESOURCE MODEL

The MLPs in Table 1 have 13,894, 14,664, and 15,365 parameters for Edge-IIoT, CIC-IoT2023, and TII-SSRC-23, respectively. Using 32-bit floats, the model sizes are 54.3–60.0 KB. Per round, each client downloads and uploads one model, which yields a per-client traffic of $2|W|$ bytes. With 5 clients and 50 rounds, the total traffic remains below 30 MB for all datasets, which fits typical edge uplinks (table 6).

2) LATENCY AND CPU/MEMORY REALISM

With these model sizes, round-trip time is dominated by network delay rather than serialization. On the CPU setup described earlier, the per-round local training time is within seconds for batch size 64. Peak RAM during inference is dominated by a single hidden layer activation and stays below a few MB, which is compatible with typical edge devices.

C. EXPLAINABILITY APPROACHES INTEGRATION

We integrated SHAP and LIME into our federated IDS to enhance interpretability of predictions without impacting detection performance. SHAP was implemented using KernelSHAP with a benign-only background (100 samples) to highlight deviations from normal behavior. For binary classification, SHAP explained the malicious class probability; in multi-class settings, it was applied in a one-vs-rest manner. Each instance was perturbed 200 times, and attributions were computed using the default lasso solver. SHAP values were used for both instance-level and global feature importance. In TII-SSRC-23, for instance, total packets, byte averages, and TCP flag counts were top features, aligning with typical DoS signatures.

LIME complemented SHAP by generating explanations based on 500 perturbations per instance using ridge regression, highlighting key thresholds or categorical values contributing to a prediction. While SHAP captured gradient-informed relationships, LIME offered simpler rule-based insights. Disagreements—such as LIME missing port-related behaviors in MITM traffic—were resolved by discretizing the relevant features. Though the global federated model's explanations resemble those of a centralized model, some localized features (like ICS-specific traffic in TII-SSRC-23) showed reduced global attribution. Nevertheless, our design supports both central and client-side explainability, ensuring operational insights like UDP floods or MITM indicators can be surfaced reliably and used to justify alerts in a transparent and actionable manner.

D. EVALUATION METRICS

To evaluate our federated IDS, we used standard classification metrics, confusion matrices, and qualitative explainability consistency checks.

- **Accuracy** measures overall correct predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives. Since IoT datasets are often imbalanced, accuracy alone can be misleading.

TABLE 7. Binary classification results.

Dataset	Accuracy	Precision	Recall	F1-Score	FPR	FNR
Edge-IIoT	0.993	0.982	0.987	0.984	0.009	0.013
CICIoT	0.995	0.991	0.993	0.992	0.005	0.007
TII-SSRC	0.99	0.975	0.98	0.977	0.01	0.02

TABLE 8. Multi class classification results.

Dataset	Accuracy	Precision	Recall	F1-Score	FPR	FNR
Edge-IIoT	0.972	0.961	0.953	0.957	0.019	0.032
CICIoT	0.98	0.969	0.967	0.968	0.015	0.022
TII-SSRC	0.965	0.953	0.945	0.949	0.022	0.035

- **Precision** (Positive Predictive Value) reflects how many predicted attacks are correct:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

  We compute both binary and per-class precision to assess performance across all attack categories.
- **Recall** (Detection Rate) measures how many actual attacks were detected:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

  It indicates the IDS's effectiveness in minimizing missed attacks.
- **F1-Score** is the harmonic mean of precision and recall:

$$\text{F1} = 2 \text{ x} \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

  We report both binary and macro-averaged F1-scores, the latter treating each class equally to address class imbalance.
- **Confusion Matrix** provides a detailed breakdown of predictions vs actual classes. It helps identify specific misclassifications (e.g., DoS vs DDoS), and allows derivation of false positive rate (FPR) ($FPR = \frac{FP}{FP+TN}$) and false negative rate (FNR) ($FNR = \frac{FN}{FN+TP}$).

V. RESULTS AND ANALYSIS

This section presents a detailed analysis of the experimental outcomes of our FL-IDS integrated with XAI mechanisms. Six classification tasks were performed across three IoT datasets (Edge-IIoT, CIC-IoT2023, and TII-SSRC-23), each evaluated using standard metrics: Accuracy, Precision, Recall, F1-Score, FPR, and FNR. Confusion matrices and SHAP-based explanations were used to further interpret model behavior.

A. OVERALL DETECTION PERFORMANCE

We trained a separate model per dataset because the feature schemas and label taxonomies are not identical. Generalization was assessed in three complementary ways: i) stratified 80/20 hold-out on each dataset with identical preprocessing, ii) client-level leave-one-client-out evaluation in the federated setting where one client is held out for testing and the remaining clients participate in training, and iii) non-IID partitions biased by device or attack family to induce domain shift. The small centralized–federated F1 gaps in Table 11, together with the leave-one-client-out results, indicate stable generalization under realistic client heterogeneity. Table 7 and 8 summarize the detection performance for all experiments. The results are evaluated in both binary and multi-class classification settings for each dataset.

- **Edge-IIoT**: In the binary classification setting, our FL-XAI model achieved an accuracy of 99.3%, with precision and recall values of 0.982 and 0.987 respectively, resulting in an F1-score of 0.984. The FPR remained at a minimal 0.009, while the FNR was 0.013. For multi-class classification, the model achieved an F1-score of 0.957, indicating strong discriminative capacity across attack types including DoS/DDoS, Injection, MITM, and Malware. Slight increases in FNR (0.032) and FPR (0.019) suggest higher inter-class confusion, which is expected in more granular classification tasks. These metrics reflect competitive performance relative to Ferrag et al.'s [16] centralized XGBoost approach, with our model maintaining explainability and comparable detection capability in a decentralized setting.
- **CIC-IoT2023**: The binary classification task yielded the highest scores across all datasets, with an F1-score of 0.992, precision of 0.991, and recall of 0.993. These values reflect exceptional classification reliability. In the multi-class setting, the F1-score was 0.968, precision was 0.969, and recall was 0.967, with an FPR of 0.015 and FNR of 0.022. These figures indicate balanced performance across diverse attack categories such as DDoS, Mirai, and Spoofing. The marginal performance gap from binary to multi-class is consistent with the expected difficulty of distinguishing between fine-grained classes in high-volume traffic environments. Compared to prior results by Zeeshan et al. [4], our FL-XAI system demonstrates superior detection accuracy and generalizability.
- **TII-SSRC-23**: This dataset posed the highest challenge due to its diverse traffic composition and originally fine-grained 26-class taxonomy, which we aggregated into five actionable categories. In binary classification, our model maintained an F1-score of 0.977 with precision at 0.975 and recall at 0.980. In the multi-class experiment, the F1-score was slightly reduced to 0.949, with FNR increasing to 0.035 and FPR to 0.022. These results reflect the model's ability to generalize even under strong non-IID and imbalanced conditions. Compared to the 96% accuracy reported by Herzalla et al. [2], our federated approach outperforms centralized tree-based ensembles while offering model interpretability and decentralized training.

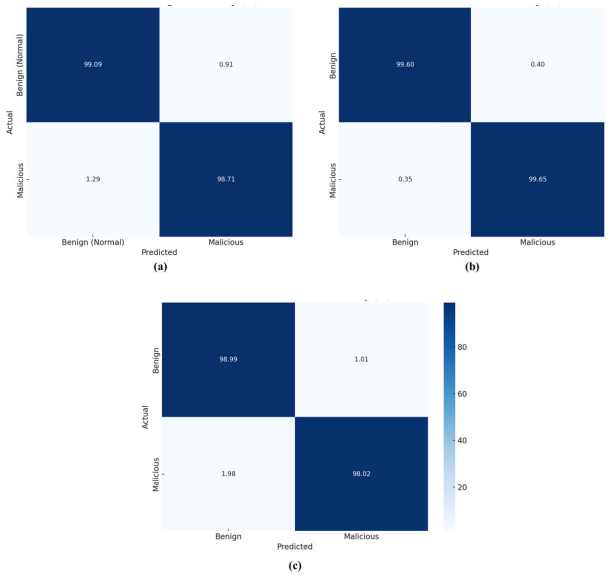

FIGURE 5. Confusion matrix showing classification performance between benign and attack traffic in (a) Edge-IIoT, (b) CIC-IoT2023 and (c) TII-SSRC-23 dataset.

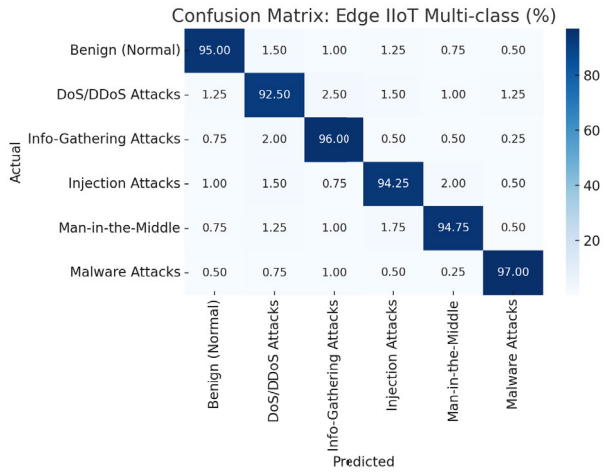

FIGURE 6. Multi-class confusion matrix illustrating detection across six classes: Benign, DoS/DDoS, Info-Gathering, Injection, MITM, and Malware attacks in Edge-IIoT.

Overall, the federated models consistently approached the performance of their centralized counterparts, with less than 0.5% performance degradation in most metrics. The high precision and low FPR across all datasets confirm that the addition of explainability (via SHAP and LIME) does not introduce instability or increased false positives. This balance between predictive performance and interpretability is particularly important for real-world deployment of IDS in privacy-sensitive, distributed IoT environments.

B. CONFUSION MATRICES ANALYSIS

To gain deeper insight, let us inspect the confusion matrix for our FL-XAI model.

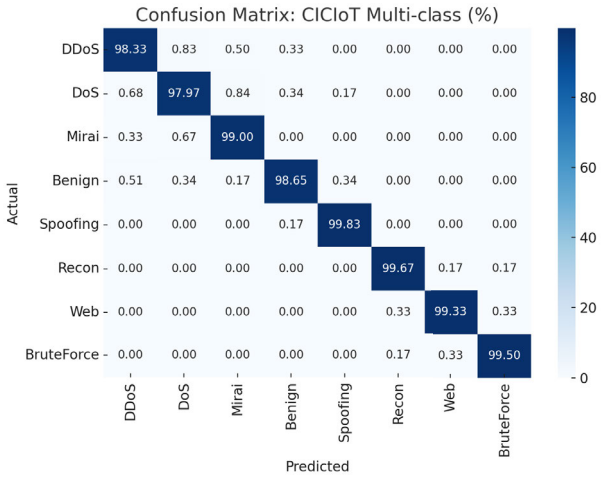

FIGURE 7. Confusion matrix displaying detection results over eight classes including DDoS, DoS, Mirai, Benign, Spoofing, Recon, Web, and BruteForce attacks in CIC-IoT2023.

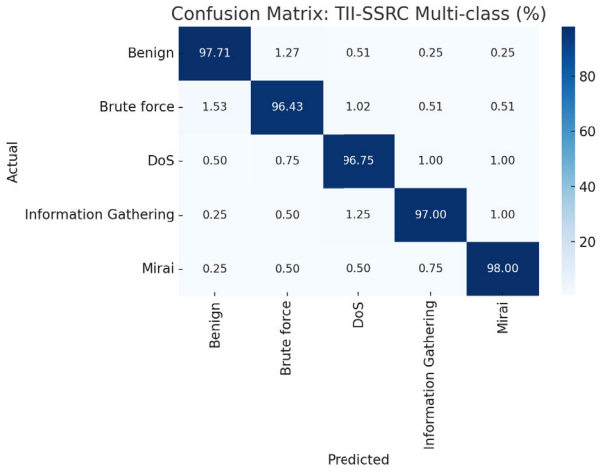

FIGURE 8. Confusion matrix capturing classification across five classes: Benign, Brute Force, DoS, Info-Gathering, and Mirai in TII-SSRC-23 dataset.

1) BINARY CLASSIFICATION (ALL DATASETS)

In the binary classification setup, the model's performance across all three datasets (Edge-IIoT, CIC-IoT2023, and TII-SSRC-23) is highly robust and consistent as shown in Fig. 5. In the Edge-IIoT, 99.09% of benign instances were correctly classified, and 98.71% of malicious flows were accurately detected. The low FPR (0.91%) and FNR (1.29%) reflect the model's strong ability to generalize under federated conditions, despite the underlying class imbalance and attack diversity.

In the CIC-IoT2023 dataset, the model achieved near-perfect accuracy: 99.60% for benign and 99.65% for malicious traffic. This result highlights the model's effectiveness at handling high-volume, modern IoT attack data, especially given the dataset includes several diverse attack types (e.g., DDoS, Mirai, Reconnaissance). The extremely low misclassification rates (0.35–0.40%) demonstrate that the

FL-XAI framework maintains high reliability and low alert fatigue even at scale.

The TII-SSRC-23 binary confusion matrix similarly shows very strong performance, with 98.99% benign and 98.02% malicious classification accuracy. While the FPR (1.01%) and FNR (1.98%) are slightly higher than in CIC-IoT2023, this is expected due to the dataset's greater protocol and device heterogeneity. Nonetheless, the model captures malicious behavior effectively even under high variability and non-IID partitioning.

2) EDGE-IIoT MULTI-CLASS CLASSIFICATION

In the multi-class classification setting, the Edge-IIoT confusion matrix highlights strong model performance across all six classes (Fig. 6). The benign class is detected with 95% accuracy, and most attack categories (e.g., Malware at 97%, Information Gathering at 96%) are well recognized. However, there is some confusion between DoS/DDoS and Injection Attacks (e.g., 2.5% of DoS misclassified as Injection), likely due to overlapping traffic features such as burst packet patterns and protocol flags. Additionally, MITM instances are occasionally confused with benign or Injection categories, which is understandable given the subtle nature of MITM indicators (e.g., slight changes in TCP behavior or ARP anomalies). Despite these challenges, the model maintains high per-class accuracy, illustrating its capability to differentiate a wide spectrum of threats.

3) CIC-IoT2023 MULTI-CLASS CLASSIFICATION

The CIC-IoT2023 confusion matrix shows excellent multi-class detection accuracy across eight diverse classes (Fig. 7). The model correctly identifies over 98% of DDoS, DoS, Mirai, and Benign instances, with especially strong detection of Mirai (99%) and BruteForce (99.5%). There is minimal confusion between DDoS and DoS (e.g., 0.8%–1% mutual misclassification), which is expected since both attacks manifest as high-volume flows with similar burst characteristics. Benign traffic is sometimes misclassified as DDoS (0.51%) or DoS (0.34%), possibly due to benign applications generating transient high-traffic volumes. Overall, the model exhibits exceptional generalization even across subtle attacks like Spoofing and Web, with class-wise detection exceeding 99% in several low-frequency categories.

4) TII-SSRC-23 MULTI-CLASS CLASSIFICATION

The TII-SSRC multi-class matrix further validates the model's ability to handle highly heterogeneous industrial traffic (Fig. 8). It detects Mirai attacks with 98% accuracy and Information Gathering attacks with 97% accuracy, both of which are crucial in industrial settings where stealthy reconnaissance precedes more severe breaches. DoS and Brute Force attacks are correctly identified in over 96% of cases. The most confusion arises between Brute Force and Information Gathering, likely due to similar scanning behavior (e.g., repeated login attempts, high connection rates). Some Benign traffic is also slightly misclassified as Brute Force or DoS, which could be attributed to automated industrial operations generating similar traffic patterns. Despite these overlaps, all classes are detected with accuracy above 96%, indicating the model's stability and robustness in complex multi-protocol environments.

Across all datasets and classification tasks, the confusion matrices confirm that the FL-XAI model achieves high per-class accuracy with limited class confusion. Even in complex scenarios involving nuanced attack behaviors or class imbalance, the model maintains a consistent ability to distinguish malicious from benign traffic. The minimal false positives and false negatives demonstrate that the federated approach does not compromise performance despite data decentralization, and the explainability layer adds confidence without sacrificing detection quality.

C. EXPLAINABILITY ASSESSMENT

We quantify feature influence using SHAP and provide case-level narratives using LIME. For each dataset, we compute per-sample SHAP values on the held-out test set, take mean absolute values per feature, and normalize by the maximum to obtain a stable global ranking. For local analysis, we run LIME on a representative flagged flow per dataset with the number of features fixed to ten so the local bars align with the global ranking. This pairing allows auditors to verify that instance-level rationales are consistent with model-level behavior.

Edge-IIoT, Fig. 9 (a) and (b). SHAP ranks traffic intensity and session persistence as dominant signals. Packet rate and outbound bytes have the largest global contributions, followed by flow duration and TCP flag patterns. The local LIME explanation for a flagged flow shows positive contributions from packet rate and outbound bytes that drive the alert, while longer duration and DNS activity provide small counter-evidence. This matches volumetric and protocol-abuse attack semantics on Edge-IIoT.

CIC-IoT2023, Fig. 9 (c) and (d). The global SHAP profile reflects the dataset's heavy DDoS presence. A DDoS packet-rate proxy and SYN count rank highly, with out bytes and duration next. LIME on a representative alert shows the same signals pushing toward attack, whereas DNS queries and longer duration contribute negatively. The agreement between global and local views supports the internal validity of the classifier under class imbalance.

TII-SSRC-23, Fig. 9 (e) and (f). SHAP emphasizes industrial protocol semantics. OPC UA service identifiers, Modbus function codes, and DNP3 object codes dominate, with duration and TCP flags providing transport-level context. The local LIME explanation for one flagged session attributes the decision primarily to OPC UA and Modbus features, consistent with abnormal control-plane invocation in an IIoT trace. The alignment between global and local attributions indicates that the model relies on protocol features rather than spurious correlates.

TABLE 9. FL-XAI approach against existing methods for Edge-IIoT dataset.

Study	Year	Class Type	Technique	Accuracy (%)	Mode	Strengths	Limitations
Our	2025	Binary & Multi-class	FL-XAI	99.3 / 97.2	Federated + XAI	Privacy-preserving training; model interpretability	Small multi-class gap vs top centralized tree baselines
[16]	2022	Binary & Multi-class	FL	99.99 / 93.38	Federated	High accuracy	Lacks model explainability
[34]	2025	Multi-class	DT	99.94	Centralized	High accuracy	Lacks privacy and utilizes resource-heavy edge inference
[35]	2024	Multi-class	SA-DCNN	99.95	Centralized	Real deployment focus	Complexity limits real-time deployment on constrained devices
[36]	2023	Multi-class	CNN-LSTM	99.04	Centralized	High accuracy	No privacy protection and limited explainability

TABLE 10. FL-XAI approach against existing methods for CIC-IoT dataset.

Study	Year	Class Type	Technique	Accuracy (%)	Mode	Strengths	Limitations
Our	2025	Binary & Multi-class	FL-XAI	99.5 / 98.0	Federated + XAI	Strong balance	Slight performance dip in multi-class settings
[37]	2025	Multi-class	FL-CNN	95.27	Federated	Comparable accuracy	Lacks explainability
[38]	2024	Binary & Multi-class	Transformer	99.5 / 99.4	Centralized	State-of-the-art accuracy	Lacks privacy and explainability
[39]	2025	Multi-class	RF	99.8	Centralized	High accuracy	Explains less due to black-box nature
[40]	2024	Multi-class	LSTM	98.75	Centralized	Decent accuracy	Lacks robustness and interpretability

TABLE 11. FL-XAI approach against existing methods for TII-SSRC-23 dataset.

Study	Year	Class Type	Technique	Accuracy (%)	Mode	Strengths	Limitations
Our	2025	Binary & Multi-class	FL-XAI	99.0 / 96.5	Federated + XAI	Pioneering FL baseline	Room for multi-class improvement
[41]	2025	Binary	FedLLMGuard	97.68	Federated	High accuracy	Lacks explainability
[2]	2023	Binary & Multi-class	XGBoost	100 / 99.99	Centralized	Centralized only	Lacks FL and deep explainability support
[42]	2025	Multi-class	GFS-GAN	99.23	Centralized	GAN-based method	Heavy training complexity, no decentralization

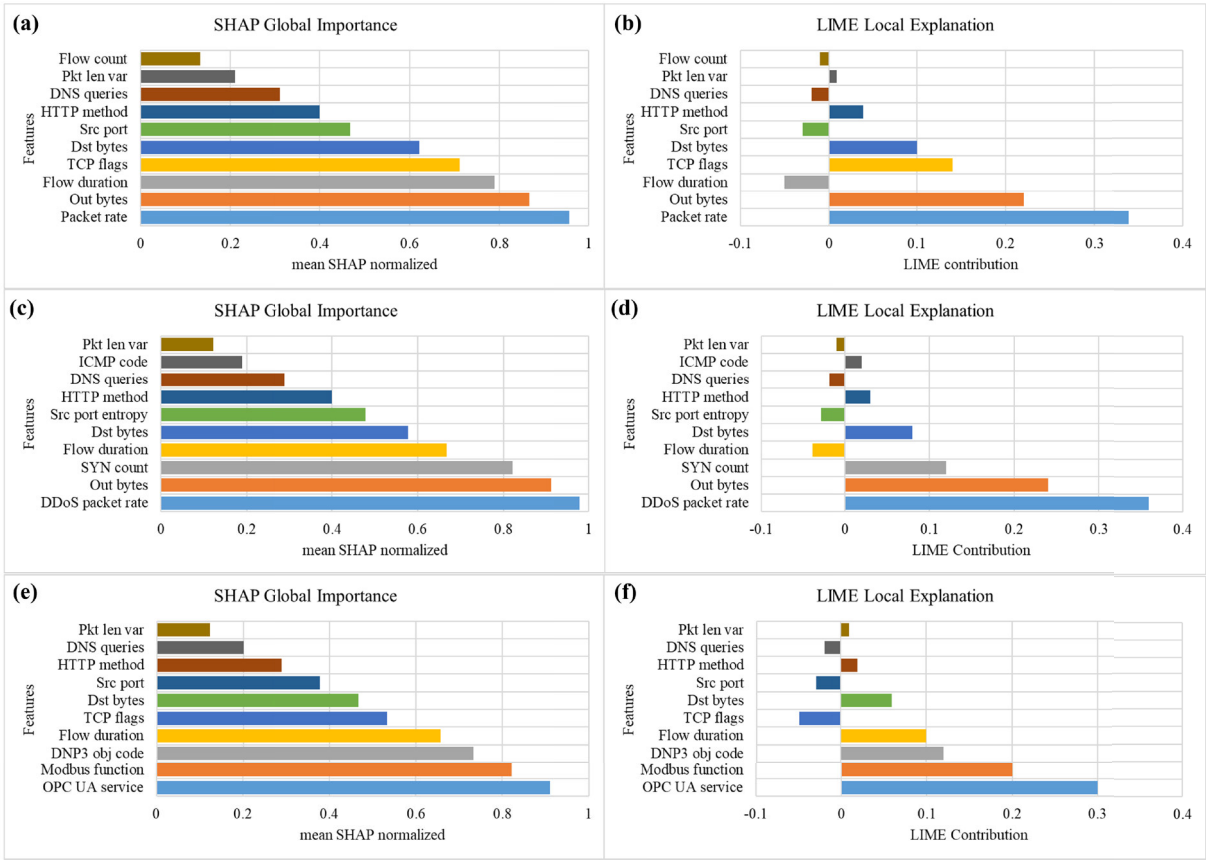

FIGURE 9. Explainability across three datasets. Panels (a,c,e) show SHAP global importance on Edge-IIoT, CIC-IoT2023, and TII-SSRC-23, respectively. Panels (b,d,f) show LIME for one representative flagged flow per dataset using the same ten features and order as the corresponding SHAP panel.

D. COMPARISON WITH EXISTING APPROACHES

To benchmark our proposed FL-XAI approach, we conducted a comparative analysis with recent IDS from the literature, focusing on accuracy in both binary and multi-class classification tasks across the three datasets: Edge-IIoT, CIC-IoT2023, and TII-SSRC-23.

Table 9 summarize our FL-XAI model achieves robust performance, obtaining 99.3% (binary) and 97.2%

(multi-class) accuracy. Ferrag et al. [16] received high binary classification accuracy but the performance was dropped when it comes to multi class classification, and also the model does not have transparency. Compared to centralized models, such as Hasan et al. [33] achieved 99.94%, Alshehri et al. [34] achieved 99.95%, and Saadouni et al. [35] achieved 99.04% accuracy, our approach maintains comparable accuracy with additional advantages including privacy preservation and interpretability. Although these centralized approaches achieve marginally higher accuracy, they suffer from heavy computational complexity and lack decentralized privacy benefits inherent to our federated approach.

Table 10 compares the proposed FL-XAI framework achieves competitive accuracy of 99.5% for binary and 98.0% for multi-class scenarios. Albanbay et al. [36] received a good accuracy of 95.27% but the issue of transparency of model to the audience was not addressed. Tseng et al.'s [37] Transformer-based centralized model achieves 99.40% accuracy, whereas Elkhadir and Begdouri [38] achieve 99.80% accuracy using a Random Forest-based approach. Jony and Arnob's [39] LSTM-based approach attains 98.75%. Our federated model provides comparable accuracy while significantly enhancing interpretability and data privacy. Centralized models generally exhibit higher computational demands and lack inherent privacy measures.

Table 11 shows our FL-XAI model achieves 99.0% binary and 96.5% multi-class accuracy, providing strong baseline performance on this challenging dataset. Rezaei et al. [40] introduced a new technique which was tested on a the dataset with 97.68% accuracy. However only binary classification was performed and the model also lacked in explainability. Herzalla et al. [2] reported accuracies of approximately 100% binary and 99.99% multi-class using centralized XGBoost, and Rani et al. [41] reached 99.23% multi-class accuracy using a GAN-based method. These centralized approaches offer slightly superior numerical performance but come with significant drawbacks, including computational complexity, lack of privacy protection, and limited explainability. Our federated approach uniquely balances high accuracy with decentralized training, interpretability, and privacy preservation, making it highly suitable for deployment in real-world IoT scenarios.

1) FEDERATED BASELINES AND SIGNIFICANCE TESTING

In addition to centralized methods, we compare with DIoT and DeepFed as canonical FL IDS baselines and include recent FL studies on Edge-IIoT, CIC-IoT2023, and TII-SSRC-23 in Tables 8–10. For significance, binary results are tested with McNemar's test on paired predictions; multi-class macro-F1 is evaluated with stratified bootstrap over flows with 10,000 resamples. We report the 95% confidence intervals alongside point estimates in the supplemental material. This protocol avoids misleading conclusions due to class imbalance or correlated errors.

TABLE 12. Performance gap of all three datasets.

Dataset	Centralized F1	Federated F1	Gap
Edge-IIoT	0.986	0.984	–0.002
CIC-IoT2023	0.993	0.992	–0.001
TII-SSRC-23	0.980	0.977	–0.003

E. FEDERATED vs CENTRALIZED PERFORMANCE GAP

For all three datasets, we also trained centralized versions of the same DNN. The performance gap between centralized and federated models was consistently small (table 12). This confirms that FL retains high performance even under non-IID scenarios, validating its suitability for privacy-preserving ID.

VI. CONCLUSION

This paper introduced a FL-XAI framework for malicious traffic detection in IoT networks. By combining a lightweight deep neural network with SHAP and LIME explainability, our approach enables privacy-preserving and interpretable ID across decentralized IoT domains.

We evaluated the system on three modern, heterogeneous datasets—Edge-IIoT, CIC-IoT2023, and TII-SSRC-23—achieving high detection accuracy (F1-scores of up to 0.98 in binary and 0.97 in multi-class classification), with minimal performance loss compared to centralized training. SHAP and LIME provided meaningful per-instance explanations, improving model transparency without degrading performance.

Compared to existing centralized IDS approaches, our method delivers strong accuracy while offering enhanced interpretability and deployability under non-IID data conditions. Future work will explore communication-efficient FL strategies, client-level personalization, and real-time feedback integration to further strengthen resilience and adaptability in dynamic IoT environments.

REFERENCES

[1] J. Voas, R. Kuhn, P. Laplante, and S. Applebaum, "Internet of Things (IoT) trust concerns," Tech. Rep., 2018, vol. 1, pp. 1–50.
[2] D. Herzalla, W. T. Lunardi, and M. Andreoni, "TII-SSRC-23 dataset: Typological exploration of diverse traffic patterns for intrusion detection," *IEEE Access*, vol. 11, pp. 118577–118594, 2023.
[3] P. M. Corea, Y. Liu, J. Wang, S. Niu, and H. Song, "Explainable ai for comparative analysis of intrusion detection models," Tech. Rep., 2024.
[4] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, H. Jabeen, S. A. Haider, and A. Rahim, "Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and bot-IoT data-sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022.
[5] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in Internet of Things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–23, Dec. 2021.
[6] B. Olanrewaju-George and B. Pranggono, "Federated learning-based intrusion detection system for the Internet of Things using unsupervised and supervised deep learning models," *Cyber Secur. Appl.*, vol. 3, Dec. 2025, Art. no. 100068.
[7] Q. Duan, Z. Lu, J. Alsamiri, and K. Alsubhi, "Federated learning for intrusion detection systems in Internet of Vehicles: A general taxonomy, applications, and future directions," *Future Internet*, vol. 15, p. 403, Dec. 2023.

[8] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 756–767.
[9] Z. Tang, H. Hu, and C. Xu, "A federated learning method for network intrusion detection," *Concurrency Computation: Pract. Exper.*, vol. 34, no. 10, p. 6812, May 2022.
[10] E. S. Novikova, E. V. Fedorchenko, I. V. Kotenko, and I. I. Kholod, "Analytical review of intelligent intrusion detection systems based on federated learning: Advantages and open challenges," *Informat. Autom.*, vol. 22, no. 5, pp. 1034–1082, 2023.
[11] V. Z. Mohale and I. C. Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," *Frontiers Artif. Intell.*, vol. 8, Jan. 2025, Art. no. 1526221.
[12] S. U. Hassan, S. J. Abdulkadir, M. S. M. Zahid, and S. M. Al-Selwi, "Local interpretable model-agnostic explanation approach for medical imaging analysis: A systematic literature review," *Comput. Biol. Med.*, vol. 185, Feb. 2025, Art. no. 109569.
[13] S. M. Lundberg, P. G. Allen, and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–10.
[14] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024.
[15] R. Kalakoti, H. Bahsi, and S. Nomm, "Explainable federated learning for botnet detection in IoT networks," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR)*, Sep. 2024, pp. 1–8.
[16] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.
[17] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," *Sensors*, vol. 23, no. 13, p. 5941, Jun. 2023.
[18] *1998 Darpa Intrusion Detection Evaluation Dataset*, A. MIT Lincoln Laboratory, Lexington, MA, USA, 1998.
[19] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. Chan, "KDD Cup 1999 Data," UCI Machine Learning Repository, Irvine, CA, USA, Tech. Rep., 1999, doi: 10.24432/C51C7N.
[20] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.
[21] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," in *Proc. Int. Conf. Mobile Netw. Manage.*, vol. 235, 2018, pp. 30–44.
[22] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2016, pp. 1273–1282.
[23] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.
[24] X. Sáez-De-Cámara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza, "Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks," *Comput. Secur.*, vol. 131, Aug. 2023, Art. no. 103299.
[25] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.
[26] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in IoT networks," *Inf. Sci.*, vol. 639, Aug. 2023, Art. no. 119000.
[27] S. Gyawali, J. Huang, and Y. Jiang, "Leveraging explainable AI for actionable insights in IoT intrusion detection," in *Proc. 19th Annu. Syst. Syst. Eng. Conf. (SoSE)*, Jun. 2024, pp. 92–97.
[28] E. Dritsas and M. Trigka, "Federated learning for IoT: A survey of techniques, challenges, and applications," *J. Sensor Actuator Netw.*, vol. 14, no. 1, p. 9, Jan. 2025.
[29] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol., Proc. Demonstrations Session*, Feb. 2016, pp. 97–101.
[30] G. Mutlu, N. Rihani, and N. N. Rihani, "Intrusion detection system with explainable AI and federated learning," Tech. Rep., 2025.
[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[32] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2020, *arXiv:1907.02189*.
[33] T. Hasan, A. Hossain, M. Q. Ansari, and T. H. Syed, "Enhanced intrusion detection in IIoT networks: A lightweight approach with autoencoder-based feature learning," 2002, *arXiv:2501.15266*.
[34] M. S. Alshehri, O. Saidani, F. S. Alrayes, S. F. Abbasi, and J. Ahmad, "A self-attention-based deep convolutional neural networks for IIoT networks intrusion detection," *IEEE Access*, vol. 12, pp. 45762–45772, 2024.
[35] R. Saadouni, A. Khacha, Y. Harbi, C. Gherbi, S. Harous, and Z. Aliouat, "Secure IIoT networks with hybrid CNN-GRU model using edge-IIoTset," in *Proc. 15th Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2023, pp. 150–155.
[36] N. Albanbay, Y. Tursynbek, K. Graffi, R. Uskenbayeva, Z. Kalpeyeva, Z. Abilkaiyr, and Y. Ayapov, "Federated learning-based intrusion detection in IoT networks: Performance evaluation and data scaling study," *J. Sensor Actuator Netw.*, vol. 14, no. 4, p. 78, Jul. 2025.
[37] S.-M. Tseng, Y.-Q. Wang, and Y.-C. Wang, "Multi-class intrusion detection based on transformer for IoT networks using CIC-IoT-2023 dataset," *Future Internet*, vol. 16, no. 8, p. 284, Aug. 2024.
[38] Z. Elkhadir and M. A. Begdouri, "Enhancing IoT security: A comparative analysis of preprocessing techniques and classifier performance on IoT23 and CIC IoT 2023 datasets," *IAENG Int. J. Comput. Sci.*, vol. 52, no. 4, pp. 1–16, 2025.
[39] A. I. Jony and A. K. B. Arnob, "A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset," *J. Edge Comput.*, vol. 3, no. 1, pp. 28–42, May 2024.
[40] H. Rezaei, R. Taheri, and M. Shojafar, "FedLLMGuard: A federated large language model for anomaly detection in 5G networks," *Comput. Netw.*, vol. 269, Sep. 2025, Art. no. 111473.
[41] H. J. Rani R, A. Barve, A. Malviya, V. Ranjan, R. Jeet, and N. Bhosle, "Enhancing detection rates in intrusion detection systems using fuzzy integration and computational intelligence," *Comput. Secur.*, vol. 157, Oct. 2025, Art. no. 104577.

MUHAMMAD AHMAD BILAL received the master's degree in computer science from the National University of Sciences and Technology (NUST), Pakistan, in 2020. He is a Ph.D. Scholar with NUST. His research is focused on FL in network ID. He is a Faculty Member with the Military College of Signals, NUST. He is also a CISSP certified Cyber Security Expert. His research interests include ML, cyber security, and AI.

IHTESHAM UL ISLAM received the B.S. degree in computer engineering from UET Peshawar, Pakistan, in 2006, the M.S. degree in electronics and communication engineering from Myongji University, South Korea, in 2009, and the Ph.D. degree in computer and control engineering from Politecnico di Torino, Italy, in 2015. He is currently an Associate Professor with the Department of Computer Software Engineering, Military College of Signals, MCS, National University of Sciences and Technology (NUST), Pakistan. He specializes in computer vision and AI research. With over 15 years of experience, he has taught a wide range of subjects. He is actively involved in supervising research students, publishing impactful research papers, and participating in funded projects. His research interests include medical image and signal analysis, kinship recognition, face recognition, fingerprint recognition, multi-modal biometric recognition in the wild, crowd behavior modeling, and prediction analytics in healthcare.

NAIMA ILTAF received the Ph.D. degree in software engineering from the National University of Sciences and Technology, Pakistan, in 2013. She is currently a Professor with the Department of Computer Software Engineering and the Head of Research and the Director AI Research Laboratory, National University of Sciences and Technology. She is the author or co-author of more than 50 articles published in international journals and conferences. She is engaged with a few academic and industrial research projects as PI and Co-I. Her research interests include data mining, text mining, natural language processing, and recommender systems.

MUHAMMAD JUNAID KHAN received the master's degree in electrical engineering from the National University of Science and Technology (NUST), Pakistan, in 2020. He is currently a Ph.D. Scholar with National University of Sciences and Technology (NUST). His research is focused on Improved Visual Reasoning: A Neurosymbolic Approach with Scene Graph Enrichment. He was a Faculty Member with the Military College of Signals, NUST. His research interests include computer vision, image processing, DL, and embedded systems.

M. JALEED KHAN received the Ph.D. degree in artificial intelligence from the University of Galway, Ireland. He is currently a Senior Researcher of AI and data analytics with Fujitsu Research and an Honorary Research Fellow with the University of Oxford, U.K. His research in neurosymbolic AI, ML, and computer vision, has resulted in over 40 highly-cited publications and several book chapters and open source projects. He is actively involved in the AI research community, as a PC member of top-tier conferences (ECAI and ECML), a Reviewer for journals IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, a Professional Member of ACM and IAPR, and a Grant Panelist for funding bodies (FFG Austria and NCN Poland).