JENRS

Received: 09 October 2026, Revised: 10 November 2026, Accepted: 14 November 2026, Online: 20 November 2026

DOI: https://dx.doi.org/10.55708/js0511006

Fed-XAI-HMAIDS: A Federated and Explainable Extension of HMAIDS for Distributed Intrusion Detection

Ola Suaifan*, Vaibhavi Tiwari*, Ramy Othman

School of Computing, Montclair State University, Montclair, NJ 07043, USA

Email(s): suaifano1@montclair.edu (O. Suaifan), tiwariv1@montclair.edu (V. Tiwari), othmanr@montclair.edu (R. Othman)

*Corresponding authors: Ola Suaifan and Vaibhavi Tiwari

ABSTRACT: Intrusion detection in distributed IoT environments is constrained by data-locality requirements, heterogeneous traffic distributions, changing attack patterns, and limited model interpretability. This work presents *Fed-XAI-HMAIDS*, an empirically evaluated federated extension of the Hierarchical Multi-Agent Intrusion Detection System (HMAIDS). The framework integrates a global One-Class Support Vector Machine (OCSVM) for anomaly screening, federated one-dimensional Convolutional Neural Network (1D-CNN) classification, sample-size-weighted FedAvg aggregation, continual learning and SHAP-based feature attribution within the HMAIDS detection–coordination–explanation architecture. Experiments on CICIoT2023 evaluate anomaly detection, federated classification, different non-IID client distributions, alternative federated configurations, repeated random-seed runs, and replay-based continual learning. The global OCSVM achieves an 83.54% detection rate with a 5.08% false alarm rate, while the three best federated configurations achieve mean accuracies ranging from 71–73% across ten repeated runs. The full pipeline achieves 71.8% accuracy in the representative training-paradigm comparison, compared with 73.0% for the corresponding FedAvg CNN baseline. SHAP analysis identifies network protocol-, flow-, and transport-related features that contribute to attack classification. The results provide empirical evidence for the feasibility of integrating anomaly screening, federated classification, continual learning, and post-hoc explainability within the HMAIDS architecture while retaining raw network traffic at the client level. The findings are limited to the evaluated dataset, four-client configuration, partition strategy, and threat model, and do not establish detection of previously unseen or zero-day attack classes.

KEYWORDS: Intrusion Detection Systems, Federated Learning, Explainable AI, Anomaly Detection

1. Introduction

Intrusion detection systems (IDS) face increasing challenges due to the scale, heterogeneity, and adversarial complexity of modern networks. Traditional machine learning (ML)-based IDS commonly depend on centralized data aggregation, creating data-exposure and communication concerns as well as centralized points of dependency [1]. Deep learning models further improve detection capability but often provide limited interpretability and remain difficult to audit in security-critical environments [2]. These limitations are especially important in Internet of Things (IoT) and critical-infrastructure deployments, where traffic is distributed, resource availability varies, and detection decisions should remain interpretable and data-local [3].

Distributed and multi-agent IDS architectures have been investigated to address these concerns. In particular, the Hierarchical Multi-Agent Intrusion Detection System (HMAIDS) [4] introduced a conceptual architecture that coordinates detection, response, and explanation through specialized agents and a central supervisor. However, the original framework did not operationalize decentralized model training or empirically evaluate its behavior under heterogeneous client distributions. This work therefore focuses on implementing and empirically evaluating a federated extension of the specific HMAIDS architecture rather than establishing a comprehensive comparison of all possible IDS deployment paradigms. The trade-offs among centralized, standalone local, and federated IDS training that motivate this extension are summarized in Figure 1; a comprehensive experimental comparison of these deployment paradigms is beyond the scope of the present study.

This paper presents *Fed-XAI-HMAIDS*, an operational federated extension of the specific HMAIDS architecture. Its novelty does not arise from the independent use of Federated Learning (FL), OCSVM, CNN, or SHAP, which have previously been studied in intrusion detection, but from their integration within the HMAIDS detection–coordination–explanation structure. The framework combines a global OCSVM anomaly-screening stage, federated multi-class CNN classification, sample-size-weighted aggregation, incremental training, and SHAP-based feature attribution while retaining raw traffic at local clients [5, 6, 7]. The main contributions are: (i) an implementation-level mapping of HMAIDS agents and layers to a distributed federated IDS; (ii) a hierarchical OCSVM–CNN detection pipeline in which a globally trained OCSVM is distributed

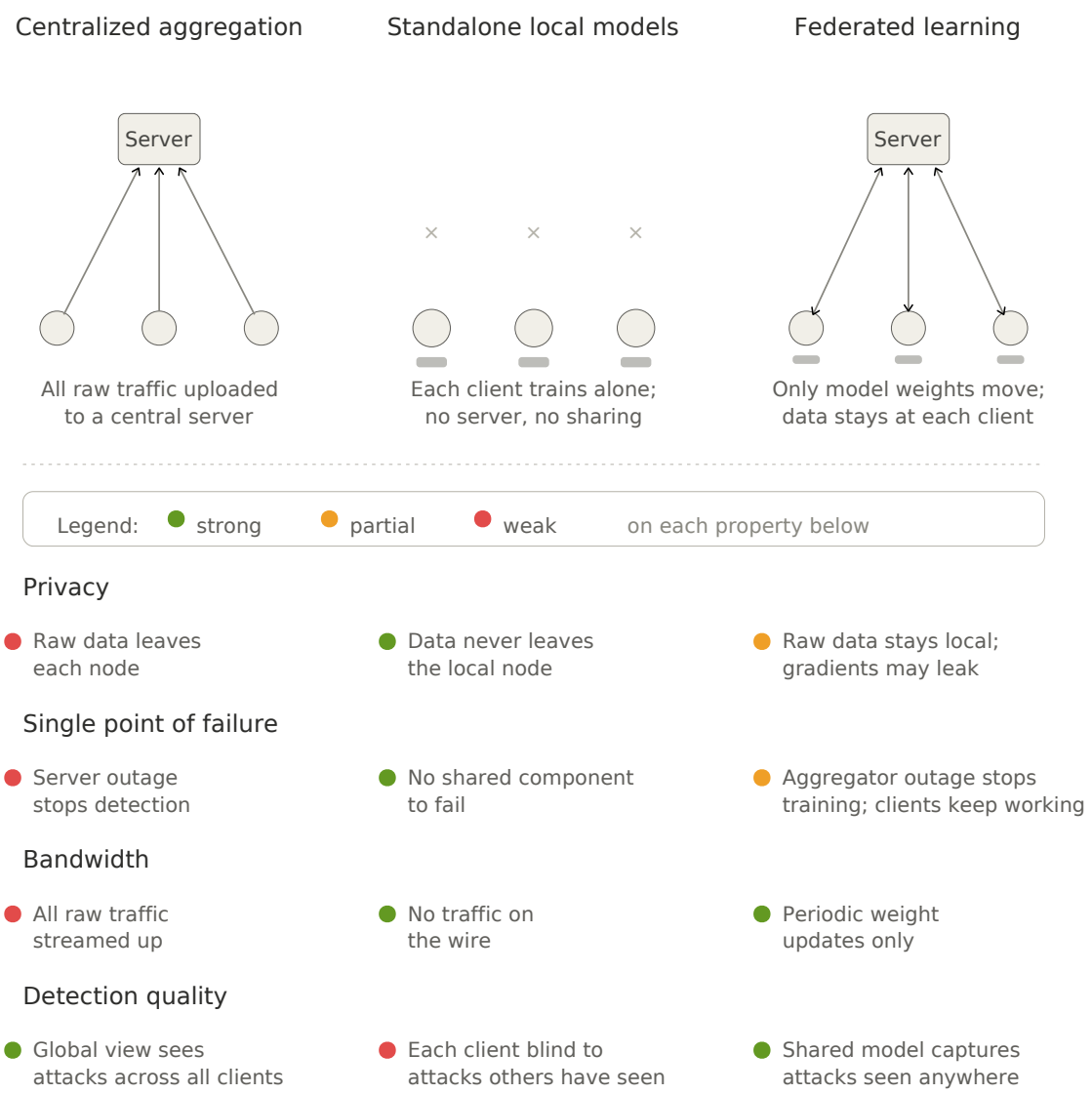


Figure 1: Comparison of centralized, standalone local, and federated IDS training across privacy, single-point-of-failure risk, bandwidth, and detection quality. Indicators denote strong (green), partial (amber), or weak (red) performance.

to clients for local anomaly screening; (iii) empirical evaluation under multiple Dirichlet-based non-IID conditions and federated configurations, including repeated-seed statistical validation; (iv) comparison of centralized, standalone, federated, and full-pipeline training paradigms; and (v) SHAP-based analysis of the traffic features influencing attack classification. The study does not claim formal privacy guarantees, large-scale deployment validation, or quantitative consistency of SHAP explanations across clients and communication rounds. To the best of our knowledge, this is the first empirical federated implementation of the specific HMAIDS framework introduced in [4]; this claim is limited to the HMAIDS framework and does not imply that the individual techniques or their broader combination are novel within federated or explainable intrusion detection.

2. Related Work

Machine learning (ML) and deep learning (DL) have been widely applied to intrusion detection. Classical approaches such as Support Vector Machines (SVM) and Random Forests (RF) provide effective classification in high-dimensional network data [8, 9], while Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks capture complex spatial and temporal traffic patterns [10]. Although these models can achieve high detection performance, they commonly require centralized data collection, introduce computational overhead, and provide limited transparency for security analysts.

Federated Learning (FL) enables distributed clients to collaboratively train a global model without transferring their raw network traffic. Existing FedAvg-based IDS frameworks demonstrate the feasibility of privacy-aware collaborative detection; however, their performance may decline when client traffic follows non-independent and identically distributed (non-IID) patterns [11]. FL also preserves data locality rather than providing a complete privacy guarantee, because shared updates may remain vulnerable to inference, poisoning, or backdoor attacks [12]. Recent federated IDS implementations have therefore investigated model aggregation, lightweight classification, attack-specific detection, and robustness under heteroge-

neous client distributions [13].

Explainable AI (XAI) has concurrently been incorporated into IDS to improve transparency and support forensic analysis. SHAP and LIME provide feature-level attribution that helps analysts understand why network flows are classified as malicious [14]. Recent studies have further combined FL and XAI for distributed intrusion detection, including SHAP-based federated models for IoT, vehicular, and malicious-traffic environments [15, 16, 17, 18, 19]. These studies establish that combining FL with post-hoc explanations is feasible; therefore, the independent integration of FL and SHAP is not presented as the novelty of this work. Moreover, post-hoc explanations introduce additional computational cost and may themselves expose information about model behavior [20].

Hybrid and hierarchical IDS frameworks provide another relevant direction by dividing detection into specialized stages. The Hierarchical Multi-Agent Intrusion Detection System (HMAIDS) [4] introduced a coordinated architecture containing detection, response and protection, explanation, and supervisory components. However, the original HMAIDS remained primarily architectural: it did not operationalize decentralized model learning or empirically validate the architecture in a federated environment. Similarly, many existing federated IDS implementations rely on a single local classifier or focus primarily on aggregation and classification accuracy rather than mapping anomaly filtering, collaborative classification, and explanation into an end-to-end hierarchical architecture.

Fed-XAI-HMAIDS addresses this narrower gap by operationalizing the specific HMAIDS architecture as a distributed intrusion detection pipeline. Unlike single-stage federated classifiers, it combines a global OCSVM, trained on benign traffic and shared across clients, with a federated 1D-CNN for multi-class attack classification and SHAP-based feature interpretation. The OCSVM uses its native decision threshold, while the CNN supports shared attack knowledge through sample-weighted federated aggregation. The framework is evaluated across multiple Dirichlet heterogeneity levels and federated configurations, followed by repeated-seed confidence-interval and significance analyses. Thus, the contribution does not lie in independently introducing OCSVM, CNN, FL, or SHAP, but in their implementation and empirical evaluation within the HMAIDS detection–coordination–explanation structure.

3. HMAIDS Overview

The Hierarchical Multi-Agent Intrusion Detection System (HMAIDS) provides a structured and adaptive framework for intrusion detection by integrating multiple intelligent agents within a unified architecture. The system is organized into three primary layers—Detection, Response and Protection, and Explanation—coordinated by a central Supervisor agent and supported by a shared Knowledge Base. This hierarchical design enables the decomposition of complex cybersecurity tasks into specialized functions, facilitating scalable, modular, and coordinated threat management [4].

The Detection layer is responsible for monitoring network activity, extracting relevant features, and identifying anomalous behavior using hybrid learning mechanisms. Once a potential threat is identified, the Response and Protection layer translates detection outcomes into actionable defense strategies through automated policy enforcement and recovery mechanisms. The Explanation layer complements this process by providing interpretable insights, forensic analysis, and human-readable reports, thereby enhancing transparency and trust in system decisions. Together, these layers form a closed-loop pipeline that supports end-to-end incident management, from initial detection to final reporting [4].

HMAIDS establishes a strong architectural foundation for coordinated, adaptive, and explainable intrusion detection in complex environments. Building upon this foundation, there exists a natural opportunity to extend the framework toward decentralized and collaborative settings, enabling scalable deployment across distributed environments while incorporating systematic empirical validation. These considerations motivate the extension proposed in this work, which extends HMAIDS into an empirically evaluated, data-local, and explainable intrusion detection pipeline.

4. Proposed Framework

This work extends the Hierarchical Multi-Agent Intrusion Detection System (HMAIDS) [4] into a decentralized, data-local, and explainable intrusion detection framework by integrating federated learning (FL) and SHAP-based explainability. The proposed system preserves the hierarchical multi-agent structure of HMAIDS while enabling collaborative model training across distributed clients without sharing raw data. This transformation evolves HMAIDS from a conceptual architecture into an implemented pipeline evaluated under heterogeneous client distributions.

4.1. Architectural Overview

The framework follows a tiered design aligned with HMAIDS, where distributed clients act as local detection agents while a central coordinator manages aggregation and orchestration. Each client independently processes local network traffic, while the central entity—analogous to the HMAIDS Supervisor—coordinates model updates through federated aggregation.

The architecture consists of three key components:

- **Local Detection Agents:** Apply the shared OCSVM for anomaly screening and train the CNN locally.
- **Federated Coordinator:** Aggregates model updates from clients using federated learning.
- **Explanation Layer:** Provides interpretable outputs using SHAP at both local and global levels.

This design preserves the HMAIDS detection–coordination–explanation paradigm while introducing decentralized intelligence and collaborative learning, as illustrated in Figure 2.

4.2. Mapping to HMAIDS

The proposed framework operationalizes key components of HMAIDS as follows:

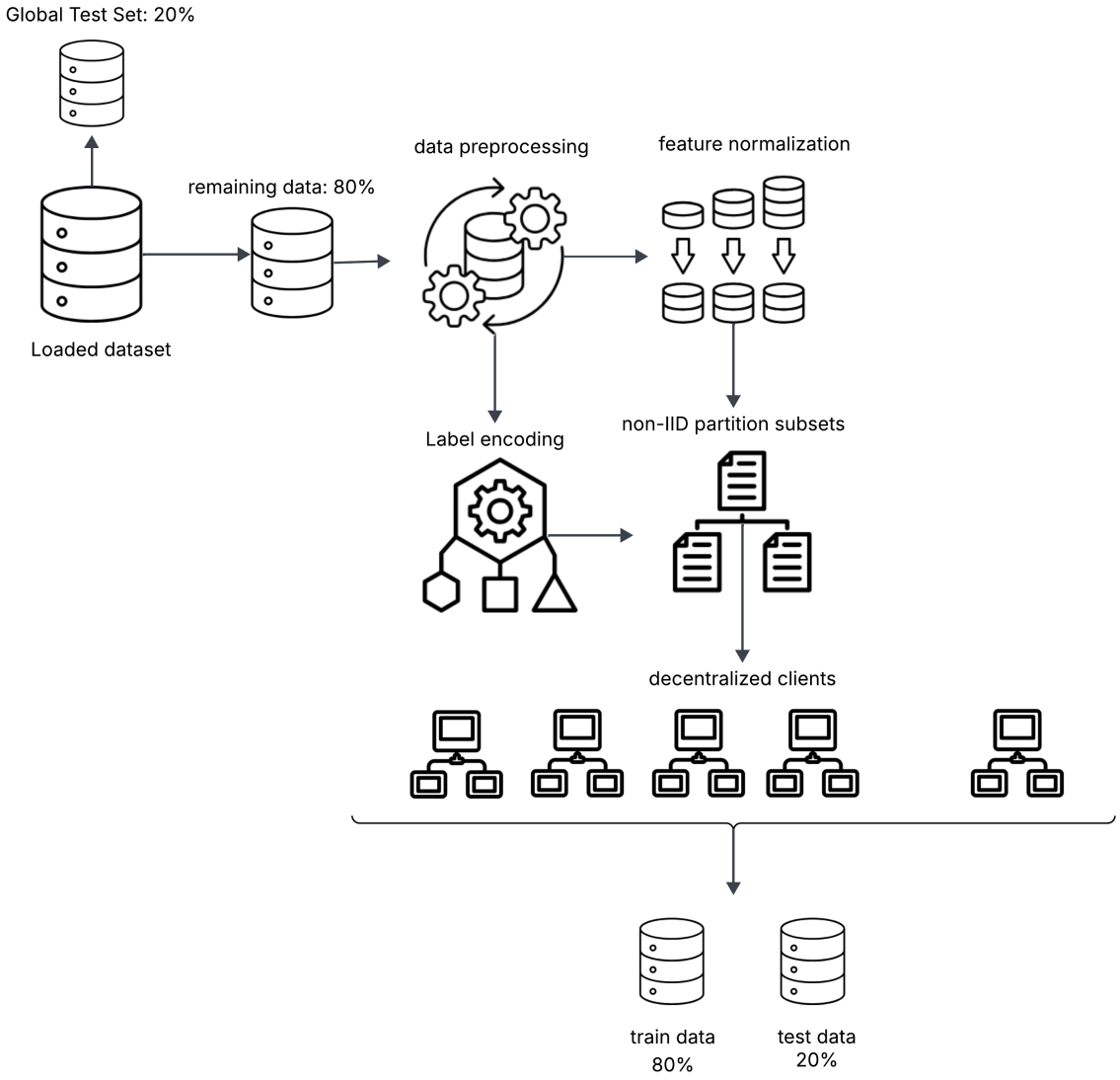


Figure 2: End-to-end architecture of the proposed federated and explainable intrusion detection framework.

- **Detection Layer:** Implemented through a hybrid OCSVM–CNN detection pipeline.
- **Supervisor Agent:** Realized as the federated coordinator responsible for aggregation and orchestration.
- **Explanation Layer:** Enhanced through SHAP-based local and global interpretability.
- **Knowledge Sharing:** Replaced by federated model updates instead of centralized data storage.

This mapping ensures architectural continuity while extending HMAIDS with practical deployment capabilities in distributed environments.

4.3. Threat Model and Assumptions

The system operates in a distributed environment in which each client observes local network traffic with potentially different statistical properties, resulting in non-IID data distributions. The adversary may launch both known and previously unseen attacks, including distributed denial-of-service, reconnaissance, scanning, and other IoT-oriented threats. However, detection of previously unseen or zero-day attack classes is not directly evaluated in the present study and is therefore treated as a threat scenario rather than an experimentally validated capability. The federated setting assumes honest-but-curious clients that follow the prescribed training protocol while potentially attempting to infer sensitive information from shared model updates. The OCSVM and the initial CNN weights are trained on a centrally available reference dataset; only subsequent CNN training is federated.

The current experimental evaluation does not consider malicious federated participants that deliberately manipulate local data or submitted model parameters. Accordingly, data-poisoning, model-poisoning, backdoor, Byzantine, and collusion-based attacks are outside the scope of the present study. Such attacks may degrade the aggregated model, bias selected predictions, or cause specific malicious traffic patterns to be classified as benign. Potential mitigation strategies include robust aggregation, validation of client updates, gradient clipping, anomaly detection over submitted updates, Byzantine-resilient optimization, and reputation-based client selection. The integration and empirical evaluation of these defenses are reserved for future

work and are discussed further in Section 6.

4.4. Data Processing and Distribution

The framework utilizes the CICIoT2023 dataset to reflect modern IoT attack scenarios. The data is partitioned into a global test set and a distributed training pool, where client datasets are generated using non-IID sampling to simulate realistic deployment conditions. Preprocessing includes removal of redundant and null records, normalization using Z-score scaling, encoding of categorical features, and feature selection using Mutual Information. Network-specific identifiers such as IP addresses and ports are removed to improve generalization. No artificial balancing is applied in order to preserve real-world class imbalance.

4.5. Hybrid Detection Mechanism

The detection process follows a two-stage pipeline that combines anomaly screening with multi-class attack classification. First, a global One-Class Support Vector Machine (OCSVM) is trained exclusively on benign traffic before the data are partitioned among clients. The trained OCSVM is then shared across clients to provide a consistent anomaly-screening mechanism while keeping raw traffic at the client level. Samples identified as anomalous are forwarded to a one-dimensional CNN for multi-class classification.

The CNN is implemented as a continual-learning network consisting of two convolutional blocks and three dense layers, with the complete configuration reported in Tables 3 and 4. Its parameters are trained collaboratively across clients using federated learning, allowing knowledge from distributed client data to be incorporated without transferring raw traffic to a central server. This staged design separates normal-traffic screening from attack classification and reduces the classification workload by forwarding only traffic identified as anomalous by the OCSVM to the CNN.

4.6. Federated Training Process

The CNN model is trained using a federated learning protocol. A global model, initialized from centrally pre-trained weights, is distributed by the coordinator to all clients. Each client performs local training on its own dataset and returns the updated model parameters, which are aggregated using the FedAvg algorithm to produce the global model. Experiments evaluate 1, 3, and 5 communication rounds (Table 5). The approach enables data-local learning across heterogeneous clients, as illustrated in Figure 3. The federated training procedure is summarized in Algorithm 1.

Algorithm 1 describes the iterative federated training process. At each communication round, the server distributes the current global CNN model to the participating clients. Each client initializes its local model with the received global parameters and trains it on its local dataset for the specified number of local epochs. The resulting local models are then returned to the server together with their corresponding training-set sizes. The server aggregates these models using sample-size-weighted FedAvg, such that clients with larger local training sets contribute proportionally more to the updated global model. This process is repeated for the specified number of communication rounds, producing the final global CNN classifier without requiring the clients to transmit their raw traffic data.

Algorithm 1: Federated Training of CNN Classifier using FedAvg

Require: Global model $W^{(0)}$, number of clients K, communication rounds R, local epochs E

for $r = 1, \ldots, R$ **do**
  Server selects participating clients $\mathcal{S}_r$
  Server broadcasts $W^{(r-1)}$ to all clients $k \in \mathcal{S}_r$
  for each client $k \in \mathcal{S}_r$ in parallel **do**
    Initialize local model $W_k^{(r)} \leftarrow W^{(r-1)}$
    Train $W_k^{(r)}$ on local dataset D_k for E local epochs
    Send $W_k^{(r)}$ and $|D_k|$ to server
  end for
  Server Aggregates the received models using sample size weighted FedAvg:
    $$W^{(r)} \leftarrow \sum_{k \in \mathcal{S}_r} \frac{|D_k|}{\sum_{j \in \mathcal{S}_r} |D_j|} W_k^{(r)}$$
end for
return Final global model $W^{(R)}$

4.7. Deployment Workflow

During deployment, the system follows a hierarchical detection pipeline:

- Incoming traffic is evaluated using the shared global OCSVM.
- If classified as normal, processing terminates.
- If anomalous, the sample is forwarded to the CNN classifier.
- The CNN assigns an attack label.
- SHAP generates an explanation for the prediction.

This workflow minimizes unnecessary invocation of computationally expensive models while ensuring accurate detection and interpretability.

The detection process is formally described in Algorithm 2.

Algorithm 2: Hierarchical Detection and Classification Pipeline

Require: Input sample x, OCSVM $f(x)$, threshold α_0, CNN $C(\cdot)$
Ensure: Predicted label y and explanation E

Compute anomaly score $s \leftarrow f(x)$
if $s \geq \alpha_0$ **then**
  $y \leftarrow$ Normal
  $E \leftarrow$ None
else
  $y \leftarrow C(x)$
  $E \leftarrow \text{SHAP}(x, C)$
end if
return (y, E)

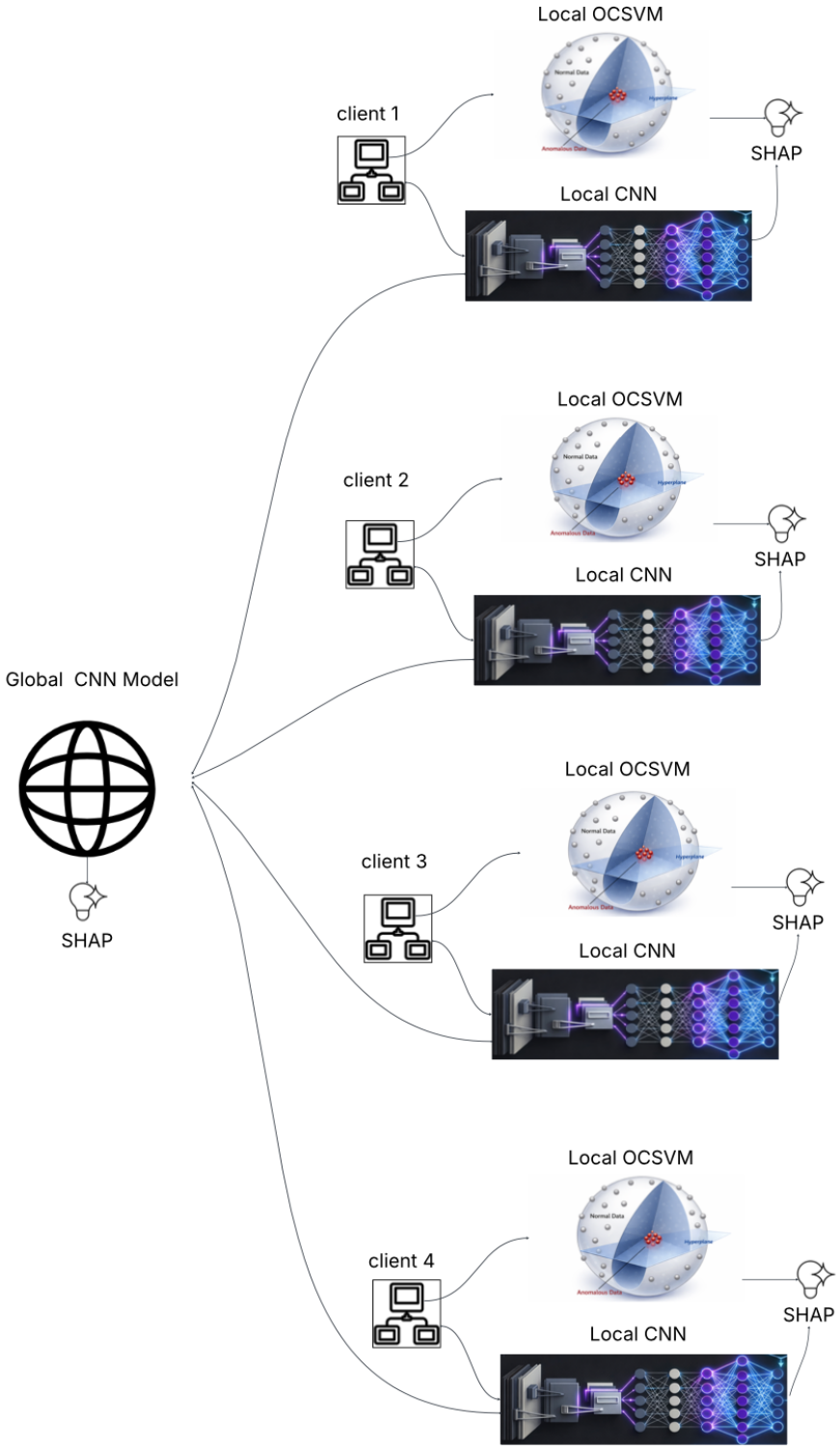


Figure 3: Federated training workflow for collaborative model learning. Each client runs a local copy of the shared global OCSVM.

4.8. Explainability Integration

Explainability is incorporated at both local and global levels using SHAP. Local explanations provide feature-level insights for individual predictions at each client, allowing the contribution of input features to a specific classification decision to be examined. Aggregated SHAP values identify the features that most strongly influence the CNN's predictions across the system. Together, these explanations provide complementary views of model behavior, supporting interpretation of individual predictions and overall feature importance. Quantitative consistency between local and global SHAP explanations across clients or communication rounds is not evaluated in the present study and is therefore considered a limitation and direction for future work.

4.9. Evaluation Framework

The framework is evaluated using detection and classification performance, robustness to heterogeneous client distributions, statistical stability across repeated runs, training-

paradigm comparison, continual-learning behavior, and model explainability. Detection performance is assessed using Accuracy, Precision, Recall, F1-score, Detection Rate (DR), False Alarm Rate (FAR), and False Negative Rate (FNR). The evaluation includes a global OCSVM anomaly-screening stage, centralized and standalone CNN configurations, federated CNN training using FedAvg, and the complete Fed-XAI-HMAIDS pipeline. Robustness to heterogeneous client distributions is examined using multiple Dirichlet concentration parameters, while repeated runs are used to assess the stability of selected federated configurations. The contribution of replay-based continual learning is evaluated through an ablation experiment. SHAP is used to analyze feature contributions to CNN predictions.

Quantitative consistency of SHAP explanations across clients or communication rounds, detailed execution-time and memory benchmarking, large-scale client evaluation, and robustness against malicious federated participants are not evaluated in the present study and are therefore treated as limitations. The evaluation consequently focuses on the empirical feasibility and predictive behavior of the proposed architecture under the specified dataset, client configuration, and threat model.

5. Experimental Evaluation

This section evaluates the proposed Fed-XAI-HMAIDS framework in terms of anomaly detection, federated attack classification, robustness under heterogeneous data distributions, statistical reliability, and explainability. The implemented pipeline consists of a global OCSVM for anomaly screening, a federated 1D-CNN for attack classification, FedAvg for collaborative learning, continual learning with replay, and SHAP for model interpretation.

5.1. Experimental Setup

5.1.1. Dataset

Experiments were conducted using the CICIoT2023 dataset, which contains benign traffic and 33 attack classes representative of real-world IoT cyberattacks [21]. To reduce computational cost while preserving the original data characteristics, representative stratified sampling was performed. Sampling quality was validated using the Chi-square goodness-of-fit test and two-sample Kolmogorov–Smirnov (KS) tests.

5.1.2. Data Preprocessing

Network identifiers (IP addresses and ports) were removed, class labels were encoded, numerical features were standardized using statistics computed only from the training set, and feature vectors were reshaped for the 1D-CNN input.

5.1.3. Federated Data Partitioning

To simulate decentralized IoT deployments, the processed dataset was partitioned among four federated clients using Dirichlet sampling. Different values of the concentration parameter α were evaluated to investigate robustness under varying degrees of non-IID data distributions.

Figure 4 illustrates the KS statistics, while Table 1 summarizes the sampled dataset.

Table 1: Summary of the CICIoT2023 dataset.

Property	Value
Dataset	CICIoT2023
Source	Canadian Institute for Cybersecurity (CIC)
Traffic type	IoT Network Traffic
Number of classes	34 (1 benign + 33 attacks)
Selected features	46
Sampling method	Stratified representative sampling
Sampling validation	Chi-square + KS tests
Chi-square statistic	2.23
Degrees of freedom	33
Chi-square p-value	1.000
KS features passing ($p > 0.05$)	46/46 (100%)

Top 10 most divergent features (for reference):

	Feature	KS_Statistic	p_value	Status
0	Radius	0.0077	0.909151	✅ Pass
1	Covariance	0.0076	0.911837	✅ Pass
2	Rate	0.0076	0.911710	✅ Pass
3	Srate	0.0076	0.911710	✅ Pass
4	IAT	0.0075	0.921331	✅ Pass
5	Variance	0.0075	0.923357	✅ Pass
6	Max	0.0072	0.944667	✅ Pass
7	flow_duration	0.0065	0.977997	✅ Pass
8	Std	0.0060	0.990705	✅ Pass
9	AVG	0.0059	0.991191	✅ Pass

Figure 4: KS: showing top 10 most divergent feature distributions

5.2. Model Configuration

5.2.1. Global OCSVM

A global One-Class Support Vector Machine with an RBF kernel was trained exclusively on benign traffic before federated partitioning because the severe class imbalance of CICIoT2023, as shown in Figure 5, provides insufficient benign samples for reliable client-level training.

Hyperparameters ν and γ were optimized through grid search. Anomaly decisions are produced using the native *scikit-learn* decision function,

$$f(x) = w^T \phi(x) - \rho. \tag{1}$$

The learned offset ρ is incorporated internally. Thus, samples with $f(x) \geq 0$ are classified as benign and those with $f(x) < 0$ as anomalous. The learned model was distributed to all clients, where it filtered anomalous traffic before CNN classification. Table 2 summarizes the configuration.

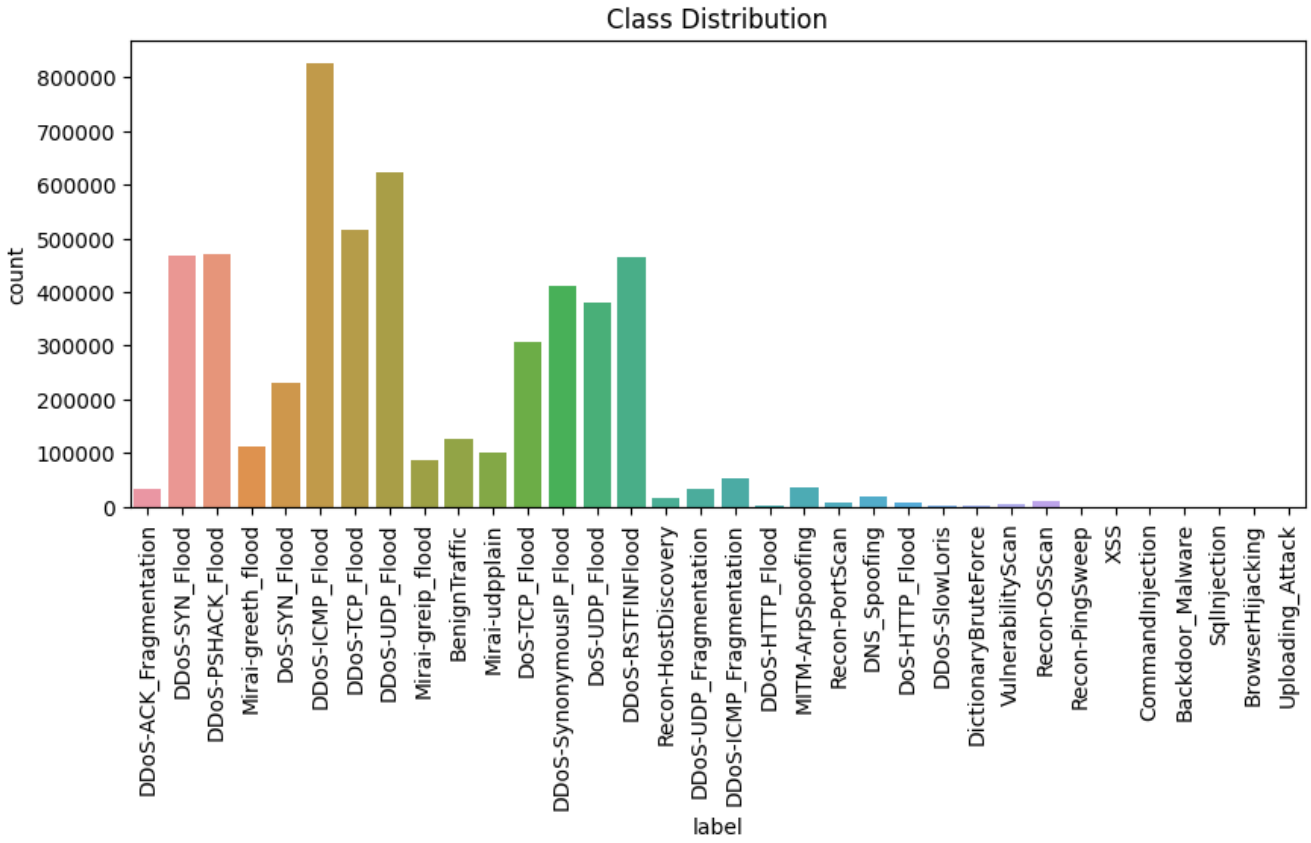


Figure 5: CICIoT2023 histogram that shows the class imbalance, particularly in comparison to the normal data.

Table 2: OCSVM configuration.

Parameter	Value
Kernel	rbf
ν (nu)	[0.05, 0.1, 0.2]
γ (gamma)	[scale, auto]
Threshold (α_0) calibration	built in sklearn ocsvm model
Decision Strategy	Native OCSVM
Parameter tuning	Grid Search
Training data	Benign traffic only
Output	Normal / Anomaly
Training strategy	Global OCSVM
Deployment	shared across all clients

The initial client-level OCSVM experiment is retained as a diagnostic comparison. Its high false alarm rates motivated the use of the global OCSVM configuration in the proposed pipeline.

5.2.2. *Federated CNN*

Detected anomalous samples were classified using a one-dimensional CNN consisting of two convolutional layers followed by fully connected layers. Continual learning was employed by progressively introducing attack classes while reusing previously learned weights. The CNN was first trained through the centralized incremental-learning procedure, after which the resulting weights were used to initialize the client CNNs. This initialization was used to retain previously learned representations and provide a common starting point for federated training. Tables 3 and 4 summarize the architecture and training parameters.

Table 3: CNN architecture used in the proposed framework.

Layer	Configuration	Kernel	Filters	Activation	Output Shape
Input	1D feature vector	–	–	–	(1, 46)
Conv1D	1D convolution with kernel size 3	Kernel size 3	8 filters	ReLU	(8, 44)
MaxPooling1D	pool size 2	–	–	–	(8, 22)
Conv1D	1D convolution with kernel size 3	Kernel size 3	16 filters	ReLU	(16, 20)
MaxPooling1D	pool size 2	–	–	–	(16, 10)
Flatten	Flatten convolution output to vector	–	–	–	(160)
Dense	Fully connected layer	–	–	ReLU	$256 + max_k = 416$ units *
Dense	Fully connected layer	–	–	ReLU	$128 + max_k = 288$ units *
Dense	Fully connected layer	–	–	ReLU	$64 + max_k = 224$ units *
Output	Softmax classification output	–	–	Softmax	34

Table 4: CNN training hyperparameters.

Parameter	Value
Framework	PyTorch 2.11.0 (CUDA 12.8)
Optimizer	Adam
Loss function	Cross-entropy
Learning rate	1×10^{-3}, 5×10^{-4}
Batch size	32
Continual phase 1 epochs	5
Continual phase 2 epochs	10
Trainable parameters	259,890
Random seed	42

Note (Table 2): $k_{\max}$ is calculated as $10k$. For the default $k = 16$, $k_{\max} = 160$.

5.2.3. Federated Learning

Federated learning was implemented using FedAvg. During each communication round, participating clients trained their CNNs locally and transmitted only model parameters for sample-size-weighted aggregation. Experiments varied the number of communication rounds, local epochs, and client participation rate. The best-performing configuration from this evaluation was subsequently used for the reported downstream experiments. Table 5 summarizes the experimental settings.

Table 5: Federated learning configuration.

Parameter	Value
Federated algorithm	FedAvg
Number of clients	4
Communication rounds	[1, 3, 5]
Local epochs	[1, 3]
Participation rate	[1.0, 0.5]
Aggregation frequency	Every communication round
Stopping criterion	None
Global model selection	Best validation accuracy

5.3. Experimental Phases and Baseline Configurations

The experiments were organized into two phases. In Phase 1, the global OCSVM and continual-learning CNN were independently evaluated to establish the performance of the two detection stages. The OCSVM-only configuration was evaluated using anomaly-detection metrics, including Detection Rate (DR), False Alarm Rate (FAR), recall, and False Negative Rate (FNR), as reported in Table 6. The CNN was subsequently evaluated under centralized and standalone settings before federated training.

Table 6: OCSVM anomaly detection performance using the native OCSVM decision function and best parameters.

Metric	Value
Detection Rate	0.8354
False Alarm Rate	0.0508
Recall	0.8381
False Negative Rate	0.1646
Parameter	**Value**
ν	0.05
γ	scale

In Phase 2, the CNN was trained using FedAvg under different communication-round, local-epoch, and client-participation configurations. The selected federated configuration was then used in the complete Fed-XAI-HMAIDS pipeline. Table 7 compares the centralized, standalone, FedAvg, and complete-pipeline configurations using the same representative data split and classification evaluation protocol. The OCSVM-only configuration is reported separately because it performs binary anomaly screening rather than multi-class attack classification.

Table 7: Comparison of training paradigms on CICIoT2023 (identical data split and metrics).

Paradigm	Accuracy
Centralized (pooled data)	0.931
Standalone (per-client, no agg.)	0.711
FedAvg (federated CNN)	0.730
Fed-XAI-HMAIDS (full pipeline)	0.718

Standalone results are averaged over four clients. Centralized serves as the pooled-data upper bound, while federated methods retain data locality. Reported values are from a single representative run.

5.4. Evaluation Metrics

Performance was evaluated using Accuracy, Precision, Recall, F1-score, Detection Rate (DR), False Alarm Rate (FAR), and False Negative Rate (FNR).

5.5. Results

5.5.1. OCSVM Anomaly Detection

Training independent OCSVMs on individual clients resulted in extremely high FAR because each client contained too few benign samples to accurately model normal traffic (Table 8).

Table 8: Initial OCSVM local training per client with poor performance due to class imbalance.

Client	FAR	Recall
0	1.0000	0.9928
1	0.5556	0.9756
2	0.8000	0.9843
3	0.4444	0.9830

The proposed global OCSVM substantially improved anomaly detection by reducing FAR to 5.08% while maintaining a Detection Rate of 83.54% (Table 6), demonstrating that training on the complete benign dataset provides a more representative normal traffic model.

The experimental results evaluate attack classes represented in the training and evaluation data under the described incremental and federated settings. They should not be interpreted as evidence of zero-day or previously unseen attack detection, since no attack class was completely withheld from training for a dedicated unseen-attack evaluation.

5.5.2. CNN Classification

Following anomaly screening, the federated CNN classified detected anomalous traffic into attack categories. The client-level results demonstrate variation across clients, reflecting differences in their local data distributions. These

results are interpreted as evidence that local CNN models can learn attack categories under the evaluated heterogeneous client configuration rather than as a direct measure of generalization to unseen deployments. Table 9 summarizes client performance, while Figures 6 and 7 present per-epoch and final per-client accuracy.

Table 9: CNN performance across federated clients.

Client	Learning rate	Accuracy	Weighted F1-score
0	0.0005	0.8525	0.8432
1	0.0010	0.9472	0.9283
2	0.0005	0.8157	0.7530
3	0.0005	0.8299	0.8059

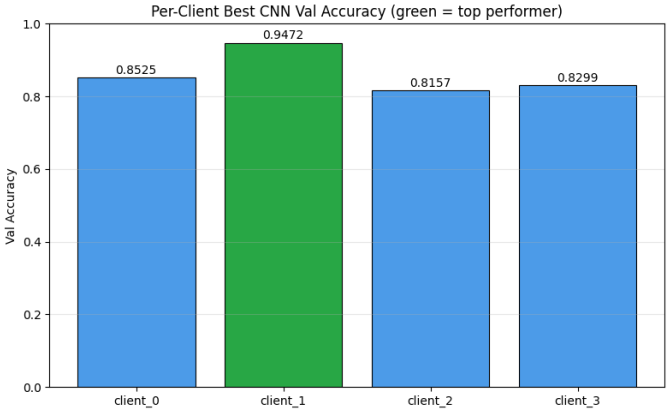


Figure 6: CNN accuracies

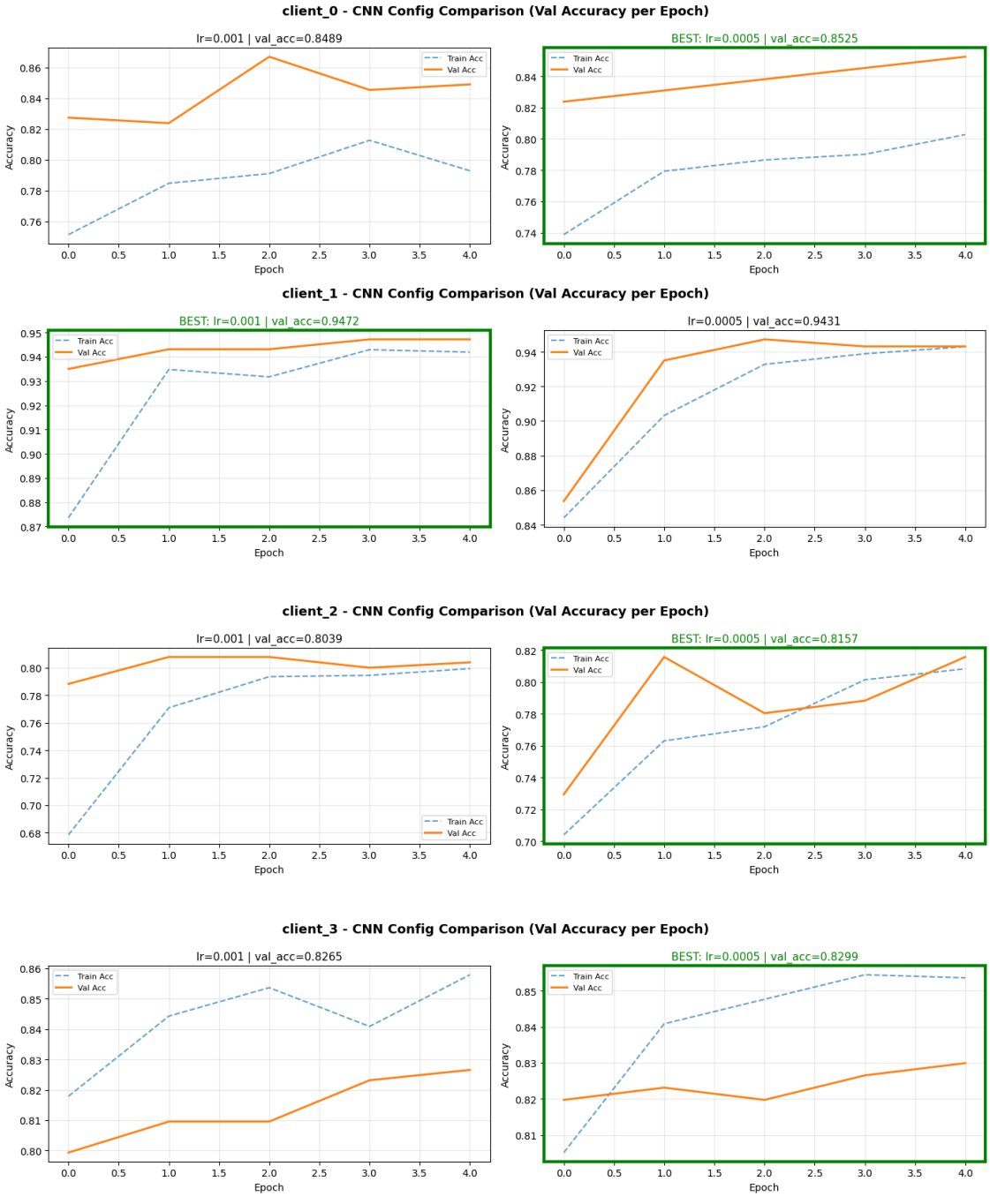


Figure 7: Per-epoch training and validation accuracy of each client CNN for two learning rates.

5.5.3. Federated Learning Performance

The federated CNN achieved competitive classification performance across the evaluated communication configurations without transferring raw network traffic. Performance varied with communication rounds, local training epochs, and client participation, demonstrating the sensitivity of federated classification to the selected training configuration (Figures 8 and 9).

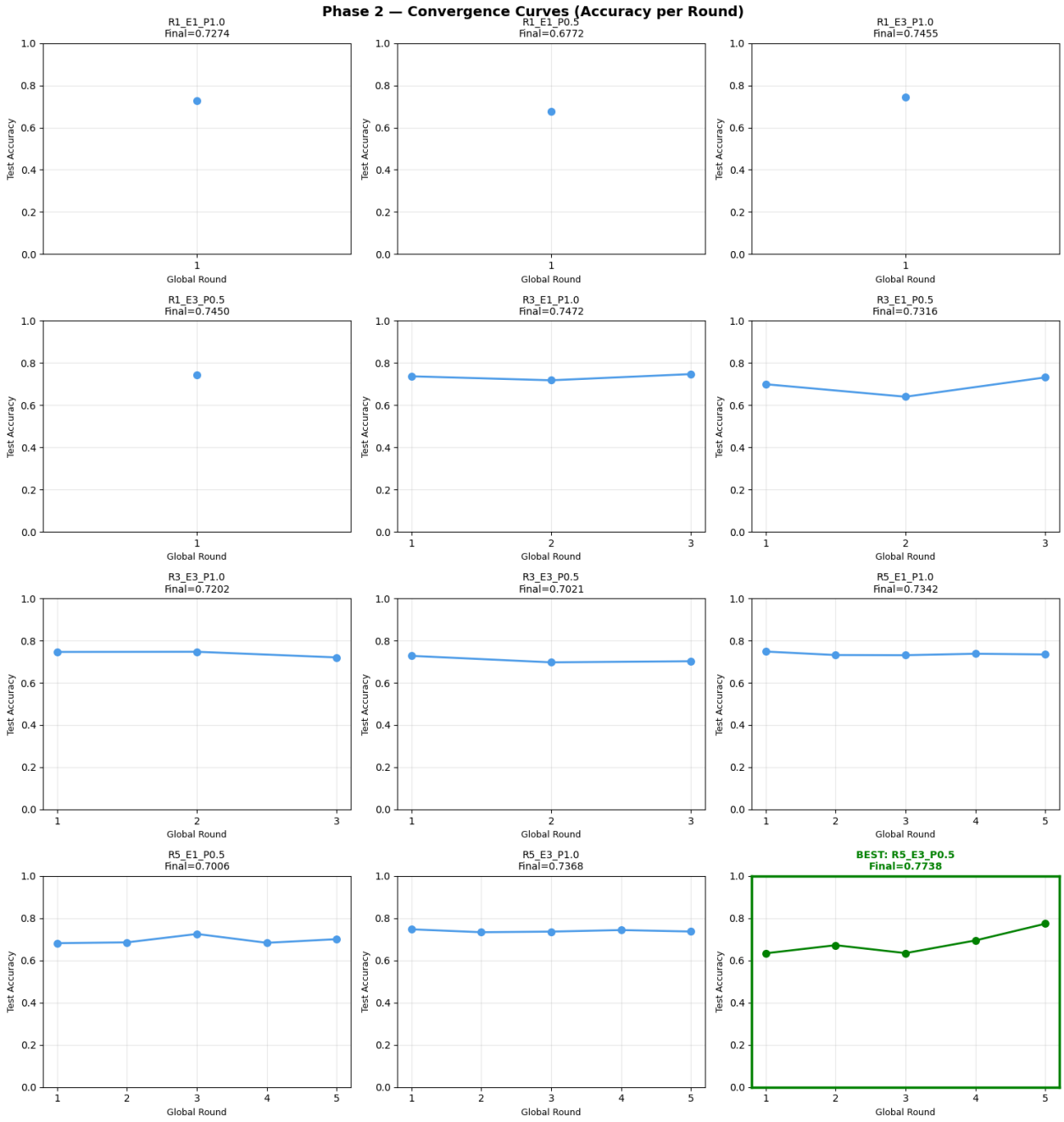


Figure 8: Accuracy convergence across communication rounds.

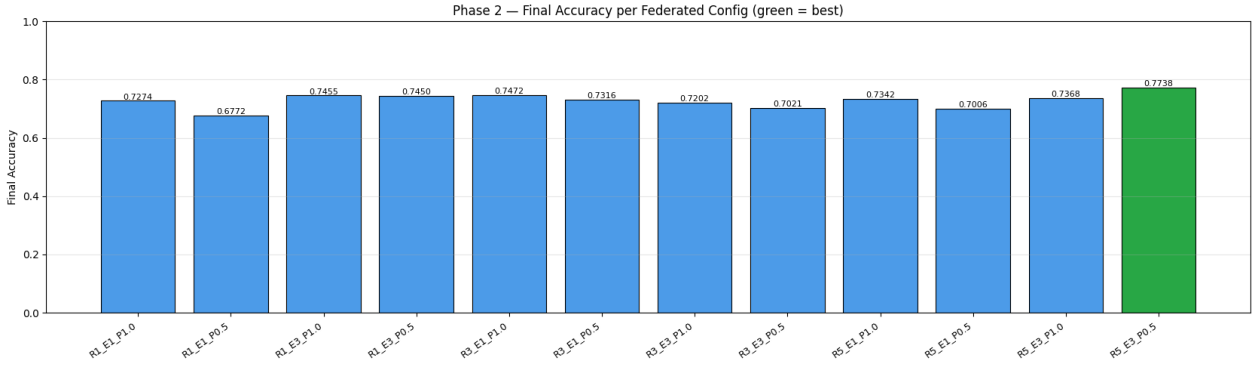


Figure 9: Final accuracy across federated learning configurations, with the best-performing configuration highlighted in green.

5.5.4. Effect of Non-IID Data

Accuracy ranged from 57.00% ($\alpha = 0.1$) to 74.22% ($\alpha = 10$), rising sharply between $\alpha = 0.1$ and $\alpha = 1$ and changing little between $\alpha = 1$ and $\alpha = 10$, which suggests that performance saturates near IID conditions. F1-score increased monotonically from 0.497 to 0.687.

Smaller α values produce highly skewed non-IID client distributions, whereas larger values generate more balanced partitions approaching IID conditions. For each setting, federated training was repeated using identical hyperparameters to isolate the effect of heterogeneity on convergence and classification performance.

Detailed results are presented in Table 10 and Figure 10.

Table 10: Influence of client heterogeneity on federated learning performance.

α	Accuracy	F1	Precision	Recall
0.01	0.6027	0.4974	0.5461	0.6027
0.10	0.5700	0.5070	0.6994	0.5700
0.50	0.6652	0.5787	0.6251	0.6652
1.00	0.7411	0.6781	0.7464	0.7411
10.00	0.7422	0.6866	0.7458	0.7422

5.5.5. Statistical Validation

The three best federated configurations were repeated ten times using different random seeds {42, 123, 456, 789, 1011, 1213, 1415, 1617, 1819, 2021}. Mean accuracy, standard deviation, confidence intervals, paired Student's t-tests, and Wilcoxon signed-rank tests confirmed that performance differences between configurations were not statistically significant ($p > 0.05$), indicating no evidence of a difference among the three configurations.

Table 11 reports their accuracy distributions and 95% confidence intervals.

Pairwise significance results are presented in Table 12.

5.5.6. Training Paradigm Comparison

To position the proposed framework against alternative training paradigms, Table 7 compares a centralized model, standalone per-client models, standard FedAvg, and the full Fed-XAI-HMAIDS pipeline under an identical data split and evaluation protocol. The centralized model, which pools all data, provides an upper bound, while the federated variants retain data locality.

The standalone configuration achieved an accuracy of 0.711, while FedAvg achieved 0.730, indicating a modest improvement of 1.9 percentage points from collaborative training compared with independently trained client models. The complete Fed-XAI-HMAIDS pipeline achieved an accuracy of 0.718. Although this result is slightly below the FedAvg baseline, it remains comparable to the standalone and federated configurations. Therefore, the results do not demonstrate that the complete pipeline improves classification accuracy over FedAvg. Instead, they demonstrate the feasibility of integrating anomaly screening, federated learning, continual learning, and SHAP-based explainability within the HMAIDS architecture while retaining raw traffic at the client level. The centralized result of 0.931 further indicates the performance gap associated with decentralized training under the evaluated experimental setting.

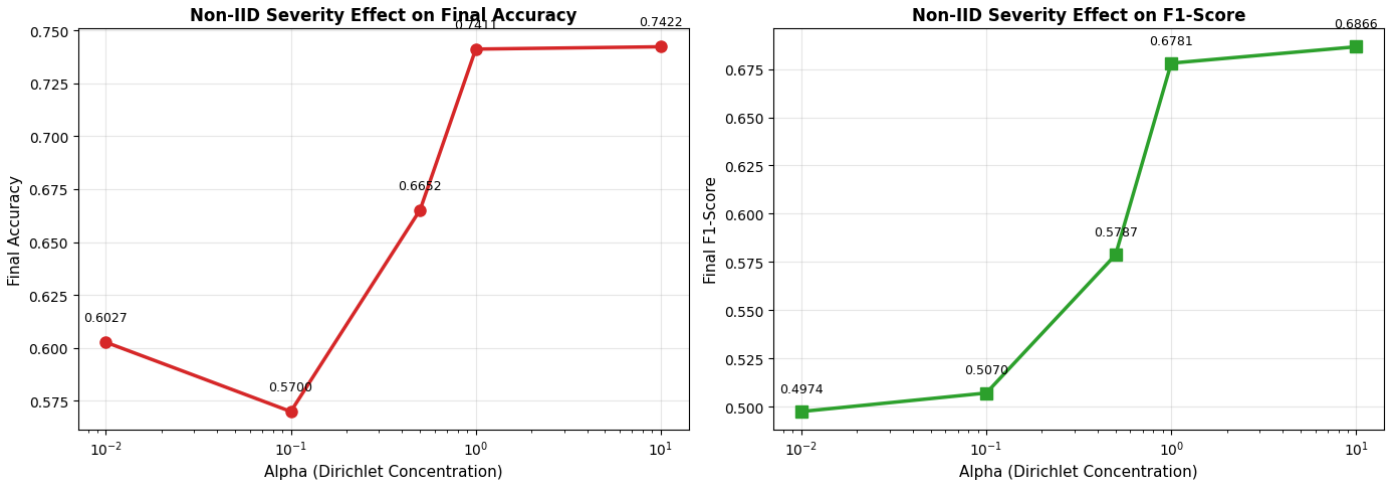


Figure 10: Effect of the Dirichlet parameter α on federated classification performance under varying non-IID conditions.

Table 11: Statistical summary of the top three federated configurations over 10 independent runs.

Config	Mean Accuracy	Std. Dev.	95% CI Lower	95% CI Upper	Min. Accuracy	Max. Accuracy
R5_E3_P0.5	0.71	0.04	0.67	0.74	0.63	0.79
R3_E1_P1.0	0.72	0.03	0.69	0.73	0.66	0.75
R1_E3_P1.0	0.73	0.02	0.71	0.74	0.68	0.75

Table 12: Pairwise statistical comparison of the top three federated configurations.

Config 1	Config 2	Mean diff.	t-statistic	t-test p-value	Wilcoxon p-value	Significant t	Significant Wilcoxon
R5_E3_P0.5	R3_E1_P1.0	-0.0049	-0.3192	0.756874	0.845703	×	×
R5_E3_P0.5	R1_E3_P1.0	-0.0168	-1.2999	0.225955	0.375000	×	×
R3_E1_P1.0	R1_E3_P1.0	-0.0120	-1.9599	0.081661	0.064453	×	×

5.5.7. Continual-Learning Ablation and SHAP Explainability

Table 13 isolates the contribution of the replay-based continual-learning mechanism. Removing replay causes a sharp collapse in accuracy, as newly introduced attack classes overwrite previously learned ones—a clear case of catastrophic forgetting. This provides direct experimental confirmation that replay-based sampling is actively used in the training pipeline and is essential rather than incidental: it materially benefits performance and prevents catastrophic forgetting.

Table 13: Ablation of the replay-based continual-learning mechanism (centralized setting, evaluated over all 34 classes).

Configuration	Accuracy	Macro-F1
With replay (proposed)	0.931	0.553
Without replay	0.023	0.003

Removing replay causes catastrophic forgetting: as new attack classes are added, previously learned classes are overwritten, collapsing overall accuracy. This confirms that replay-based sampling is essential to the continual-learning pipeline.

SHAP was used to analyze feature contributions to CNN predictions. As shown in Figure 11, the resulting global feature-importance analysis identified application-layer protocol indicators, packet-transmission characteristics, and TCP control features among the most influential attributes. Features including DNS, HTTP, HTTPS, ack_count, syn_flag_number, psh_flag_number, cwr_flag_number, ece_flag_number, Drate, and Header_Length were interpreted according to their network and transport-layer meanings. These results provide feature-level interpretability for the evaluated CNN predictions. However, the study does not quantitatively measure agreement or stability between local and global SHAP explanations across clients or communication rounds.

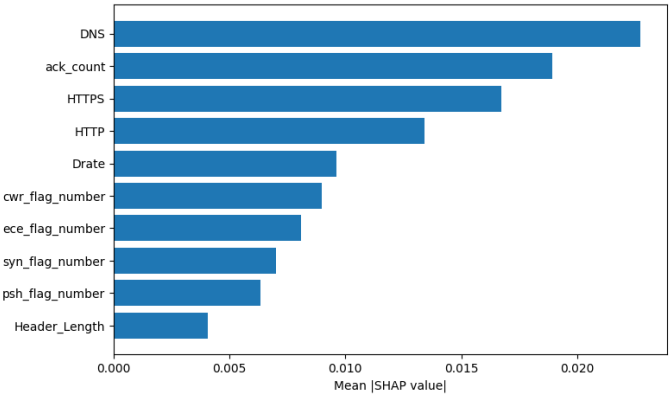


Figure 11: Global SHAP feature importance.

5.6. Discussion

The experimental findings provide initial evidence that Fed-XAI-HMAIDS can integrate anomaly screening, federated multi-class classification, and post-hoc explainability within a unified distributed intrusion detection pipeline. The results indicate that the OCSVM and CNN components perform complementary roles: the OCSVM provides an initial screening mechanism for identifying anomalous traffic, while the CNN assigns specific attack labels to samples forwarded by the anomaly detector.

The improvement observed as attack classes are incrementally added indicates that the continual-learning mechanism can incorporate additional attack categories introduced during the evaluated training process. This should not be interpreted as evidence of zero-day or previously unseen attack detection, because the present study does not include a dedicated held-out-attack experiment. Evaluation on attack classes completely excluded from training remains an important direction for future work. At the same time, the false alarm rate demonstrates that the overall performance of the hierarchical pipeline remains sensitive to OCSVM threshold calibration, because benign samples incorrectly marked as anomalous increase the downstream classification workload.

The SHAP analysis further shows that the CNN relies on interpretable application, flow, and transport-layer characteristics. Overall, the findings support the feasibility of extending HMAIDS with federated learning and SHAP-based explainability while retaining raw traffic at local client nodes. However, the results should be interpreted within the evaluated dataset, client configuration, and threat model rather than as evidence of formal privacy guarantees, unrestricted scalability, or production readiness; the broader methodological and deployment constraints are discussed in Section 6.

6. Limitations

Although the proposed Fed-XAI-HMAIDS framework demonstrates the feasibility of integrating hierarchical anomaly detection, federated classification, and explainability within a unified intrusion detection architecture, several limitations define the scope of the current evaluation.

6.1. Computational and Communication Overhead

The use of SHAP introduces additional computational overhead because feature-attribution values must be generated after model inference, and this cost may increase with traffic volume, input dimensionality, and the number of federated clients. Federated learning also requires repeated exchange of model parameters between clients and the central coordinator, with the resulting communication cost influenced by the model size, number of participating clients, aggregation frequency, and total number of communication rounds. The present study does not provide detailed empirical measurements of SHAP execution time, inference latency, training time, aggregation latency, memory consumption, or energy usage. Communication requirements are discussed in terms of model size and the number of participating clients and rounds, but a deployment-level bandwidth benchmark was not performed. These factors require further evaluation before conclusions can be drawn regarding performance on bandwidth-constrained or latency-sensitive edge devices.

6.2. Privacy Leakage and Unsupported Attack Models

The proposed federated configuration keeps raw network traffic at the client level and therefore reduces the

need for centralized raw-data collection. However, this data-locality property should not be interpreted as a formal privacy guarantee. The current implementation does not incorporate differential privacy, secure aggregation, or homomorphic encryption, and shared model parameters may still expose information about local training data. The privacy benefit of the evaluated system is therefore limited to data locality and reduced direct exposure of raw traffic. As defined in Section 4.3, the present evaluation assumes honest-but-curious clients and does not experimentally consider malicious federated participants; consequently, data-poisoning, model-poisoning, backdoor, Byzantine, and collusion-based attacks remain outside the scope of the current study. Evaluating the framework under these adversarial conditions and integrating defensive mechanisms such as robust aggregation, client-update validation, gradient clipping, anomaly detection over submitted updates, Byzantine-resilient optimization, and reputation-based client selection constitute important directions for future investigation.

6.3. Scalability and Edge-Resource Constraints

The current experiments are conducted with a limited number of simulated clients and do not evaluate larger client populations, partial participation, intermittent connectivity, or client dropout. As the scale of deployment increases, communication load, synchronization delay, aggregation efficiency, convergence behavior, and global-model stability may change. Similarly, the experiments were not conducted on resource-constrained IoT or edge hardware. Consequently, the current results do not establish large-scale scalability, deployment efficiency, or suitability for low-power hardware. Although local OCSVM screening can prevent normal traffic from reaching the CNN stage, CNN inference, federated training, and SHAP explanation generation may still impose computational requirements. Hardware-level evaluation is therefore necessary before making deployment-oriented claims.

6.4. Dataset and Generalizability

The experimental evaluation is primarily based on the CICIoT2023 dataset. Although this dataset contains diverse IoT attack categories and reflects substantial class imbalance, findings derived from a single benchmark may not directly generalize to enterprise networks, industrial control systems, cloud environments, or other IoT infrastructures. Dataset-specific characteristics may influence feature selection, anomaly thresholds, classification performance, and SHAP feature rankings. Validation using additional benchmark datasets and real-world traffic traces is therefore necessary to determine whether the observed performance is consistent across different network domains and attack distributions.

In addition, the study does not include a dedicated end-to-end evaluation of the complete OCSVM–CNN pipeline using an independently held-out test set that remains completely untouched during model, threshold, and federated-configuration selection. The existing evaluation therefore does not establish performance under such a strictly isolated test protocol. Future work should address this limitation by defining an independently held-out test set and evaluating the complete pipeline only after all model and configuration choices have been finalized. Further evaluation across multiple datasets and larger and more diverse client populations would also strengthen the assessment of generalizability.

6.5. Non-IID and Explainability Scope

The federated evaluation examines multiple degrees of heterogeneity using Dirichlet-based client partitioning. However, it does not separately evaluate all possible forms of non-IID behavior, such as independent label skew, quantity skew, and feature skew. The reported non-IID results should therefore be interpreted specifically within the evaluated Dirichlet partitioning strategy and four-client configuration.

The SHAP analysis provides feature-level explanations for the evaluated CNN predictions. However, the study does not quantitatively assess explanation consistency, agreement, or stability across clients or communication rounds. Consequently, the results demonstrate the availability and qualitative interpretation of SHAP explanations rather than establishing cross-client explanation consistency.

Overall, these limitations define the scope of the empirical evidence presented in this study. The reported findings support the feasibility of the proposed HMAIDS extension under the evaluated dataset, client configuration, training settings, and threat model, but do not establish formal privacy guarantees, unrestricted scalability, deployment efficiency, or universal generalizability.

7. Future Scope

The limitations identified in Section 6 motivate future work on class imbalance, client heterogeneity, scalability, privacy, robustness, and generalizability. Planned extensions include adaptive reweighting, focal loss, controlled sampling, broader non-IID evaluations, optimization of the OCSVM parameters ν, γ, and α_0, and ablation studies of the OCSVM, CNN, federated learning, replay sampling, and SHAP components. Further work will examine incremental learning for emerging attacks, communication and resource efficiency across larger and partially participating client populations, lightweight models for IoT and edge devices, and privacy mechanisms such as differential privacy, secure aggregation, and homomorphic encryption. The framework will also be tested against poisoning, backdoor, Byzantine, and collusion attacks using robust aggregation, update validation, gradient clipping, anomaly detection, and reputation-based client selection. Finally, future studies will assess explanation consistency across clients and rounds, compare SHAP with alternative XAI methods, and validate the framework on additional datasets, real-world traffic, and recent centralized and federated IDS baselines.

8. Conclusion

This work presented *Fed-XAI-HMAIDS*, a federated extension of the Hierarchical Multi-Agent Intrusion Detection System that integrates anomaly detection, federated

learning, continual-learning mechanisms, and SHAP-based explainability within a unified architecture. The OCSVM achieved an 83.54% detection rate with a 5.08% false alarm rate, while federated accuracy ranged from 57.00% to 74.22% depending on client heterogeneity. Across repeated runs, the selected federated configurations achieved mean accuracies of 71–73%, with no statistically significant differences. Centralized training achieved 93.1% accuracy, compared with 71.1%, 73.0%, and 71.8% for standalone, FedAvg, and full Fed-XAI-HMAIDS configurations, respectively, indicating that the framework is not intended to outperform FedAvg in classification accuracy but to integrate anomaly screening, collaborative and continual learning, and feature-level interpretability while keeping raw network traffic local. Replay ablation confirmed the importance of replay in maintaining continual-learning performance, while SHAP analysis identified relevant application-, flow-, and transport-layer features influencing CNN predictions, although explanation consistency across clients and communication rounds was not quantitatively evaluated. Overall, the study provides an initial empirical implementation of HMAIDS in a federated setting, with findings limited to CICIoT2023, a four-client configuration, Dirichlet-based partitioning, the evaluated federated settings, and an honest-but-curious threat model. Future work should investigate formal privacy mechanisms, larger-scale deployments, broader non-IID conditions, adversarial clients, cross-dataset validation, and quantitative SHAP-consistency analysis.

Funding

This research received no external funding.

Author Contributions

Conceptualization: Ola, Vaibhavi, Ramy;
Methodology: Ola;
Writing – Original Draft: Ola, Vaibhavi, Ramy;
Writing – Review & Editing: Ola, Vaibhavi, Ramy;
Visualization: Ola, Ramy.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, A. Hotho, "A survey of network-based intrusion detection data sets", *Computers & Security*, vol. 86, pp. 147–167, 2019, doi:10.1016/j.cose.2019.06.005.

[2] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead", *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019, doi:10.1038/s42256-019-0048-x.

[3] S. Sicari, A. Rizzardi, L. A. Grieco, A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead", *Computer Networks*, vol. 76, pp. 146–164, 2015, doi:10.1016/j.comnet.2014.11.008.

[4] O. Suaifan, V. Tiwari, R. Othman, K. G. Herbert, "Hmaids: A hierarchical multi-agent framework for adaptive and explainable intrusion detection", "Proceedings of the International IoT, Electronics and Mechatronics Conference (IEMTRONICS)", Imperial College London, London, United Kingdom, 2026.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data", "AISTATS", pp. 1273–1282, 2017.

[6] P. Kairouz, B. McMahan, B. Avent, *et al.*, "Advances and open problems in federated learning", *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1–210, 2021, doi:10.1561/2200000083.

[7] S. M. Lundberg, S.-I. Lee, "A unified approach to interpreting model predictions", "NeurIPS", pp. 4765–4774, 2017.

[8] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, W.-Y. Lin, "Intrusion detection by machine learning: A review", *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994–12000, 2009, doi:10.1016/j.eswa.2009.05.029.

[9] L. Breiman, "Random forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi:10.1023/A:1010933404324.

[10] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, S. Venkatraman, "Deep learning approach for intelligent intrusion detection system", *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi:10.1109/ACCESS.2019.2895334.

[11] Y. Zhao, *et al.*, "Federated learning with non-iid data", 2018, doi:10.48550/arXiv.1806.00582, arXiv preprint arXiv:1806.00582.

[12] E. Bagdasaryan, *et al.*, "How to backdoor federated learning", "AISTATS", pp. 2938–2948, 2020.

[13] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions", *Computer Communications*, vol. 195, pp. 346–361, 2022, doi:10.1016/j.comcom.2022.09.012.

[14] M. T. Ribeiro, S. Singh, C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier", "KDD", pp. 1135–1144, 2016, doi:10.1145/2939672.2939778.

[15] R. Taheri, R. Jafari, A. Gegov, F. Arabikhan, A. Ichtev, "Explainable ai for federated learning-based intrusion detection systems in connected vehicles", *Electronics*, vol. 14, no. 22, p. 4508, 2025, doi:10.3390/electronics14224508.

[16] K. Fatema, S. K. Dey, M. Anannya, R. T. Khan, M. M. Rashid, C. Su, R. Mazumder, "Federated xai ids: An explainable and safeguarding privacy approach to detect intrusion combining federated learning and shap", *Future Internet*, vol. 17, no. 6, p. 234, 2025, doi:10.3390/fi17060234.

[17] R. Kalakoti, S. Nõmm, H. Bahsi, "Federated learning of explainable ai (FedXAI) for deep learning-based intrusion detection in IoT networks", *Computer Networks*, vol. 270, p. 111479, 2025, doi:10.1016/j.comnet.2025.111479.

[18] M. A. Bilal, I. U. Islam, N. Iltaf, M. J. Khan, J. Khan, "Federated learning with explainable ai for malicious traffic detection in IoT networks", *IEEE Access*, vol. 13, pp. 173368–173383, 2025, doi:10.1109/ACCESS.2025.3613459.

[19] M. H. Gholamrezazadeh, A. Montazerolghaem, "XAI FL-IDS: A federated learning and SHAP-based explainable framework for distributed intrusion detection systems", 2026, doi:10.48550/arXiv.2605.19448, arXiv:2605.19448.

[20] D. Slack, *et al.*, "Fooling lime and shap: Adversarial attacks on post hoc explanation methods", "AAAI/ACM Conference on AI Ethics and Society", pp. 180–186, 2020, doi:10.1145/3375627.3375830.

[21] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment", *Sensors*, vol. 23, no. 13, p. 5941, 2023, doi:10.3390/s23135941.

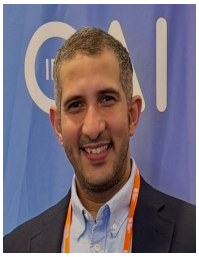
Copyright: This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-SA) license (https://creativecommons.org/licenses/by-sa/4.0/).



Ola Suaifan holds a Master's degree in Data Science from Montclair State University, a MicroMasters credential in Supply Chain Management from MIT, and a Bachelor's degree in Mathematics from the University of Jordan. She brings over eight years of experience in supply chain management, with current focus on integrating artificial intelligence into supply chain operations. Her research sits at the intersection of applied data science and data science systems development, with recent work exploring the integration of agentic AI in cybersecurity.



Vaibhavi Tiwari holds a MicroMasters credential in Data and Statistics from the Massachusetts Institute of Technology (MIT), a bachelor's degree in Computer Applications, a Master of Computer Applications from the National Institute of Technology Karnataka (NITK), Surathkal, and a Master of Science in Computer Science from Montclair State University. With over nine years of experience in healthcare technology, she specializes in data security, artificial intelligence, cloud technologies, and large-scale data management. Her research focuses on cybersecurity, healthcare informatics, artificial intelligence, cloud computing, and big data analytics. She has published in IEEE conferences, including work on ransomware recovery using digital forensics and drone data recovery. Her research emphasizes emerging cyber threats, incident response, digital forensics, data recovery, and the protection of sensitive information. Vaibhavi is also a certified Globee Awards judge and actively contributes to mentorship and community service through Women in Big Data and the International Science and Engineering Fair (ISEF), supporting students and emerging professionals in science, technology, engineering, and data-related fields.



Ramy Othman is the Assistant Director of Technology and Infrastructure at the College of Science and Mathematics, Montclair State University, where he oversees GPU cluster management, Kubernetes orchestration, JupyterHub multi-user environments, and RAG pipeline development. He holds a Bachelor's degree in Electrical, Electronics and Communications Engineering from the Faculty of Engineering, Alexandria University, Egypt, and an MSc in Data Science from Montclair State University. In addition to his infrastructure role, he served as an Adjunct Professor in the School of Computing, delivering courses in Python Programming, Data Structures and Algorithms, Systems Programming, and Computer Networks. His research interests lie at the intersection of AI/ML infrastructure, cybersecurity, and applied machine learning, with a focus on scalable, GPU-accelerated systems and multimodal AI pipelines. Othman earned First Prize in the Graduate Track of the RAISE-24 Informatics and Data Science Competition. He also built and manages the university's dedicated Network and AI research labs, through which he has mentored students.