SINTECHCOM: Science, Technology and Communication Journal
Vol. 7, No. 1, October 2026, pp. 69-82, DOI: 10.59190/stc.v7i1.424

Unsupervised federated autoencoders for IoT botnet detection under non-IID and contaminated training data

Muhammad Ikhsan[1,2,*], Imam Riadi[1], Ardiansyah Ardiansyah[1]
[1]Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta 55191, Indonesia
[2]Department of Computer Science, UIN Sumatera Utara Medan, Medan 20371, Indonesia

ABSTRACT

Federated learning (FL) has been proposed for privacy-preserving intrusion detection in internet of things (IoT) networks, yet most studies assumed labeled traffic and identically distributed client data. This study benchmarked unsupervised federated autoencoders for IoT botnet detection under non-IID heterogeneity and training-data contamination. N-BaIoT and Edge-IIoTset were partitioned into clients using IID, natural per-device, and Dirichlet partitions with concentration parameters of 1.0, 0.5, and 0.1, and botnet traffic was injected into the benign training data of 30% of the clients at rates of 0% – 20%. Lightweight autoencoders were trained on unlabeled traffic and aggregated with FedAvg and FedProx, with local-only and centralized training as references, in 264 runs over five random seeds. FedAvg did not outperform local-only training in any of the ten dataset–partition settings (mean AUC-PR difference of 0.058, Holm-adjusted p = 0.014) and, on N-BaIoT, declined by up to 0.087 after an early peak. FedProx prevented this decline on N-BaIoT (AUC-PR up to 0.999) but gave no gain on Edge-IIoTset. Under heterogeneity, a single global threshold raised the false positive rate from about 1% to 22.2%, whereas client-level thresholds kept it at or below 1.0%. Contamination of 20% reduced the worst-client F1-score by 0.33 for FedAvg and 0.47 for FedProx, and client-level thresholds did not recover this loss. These findings indicated that standard aggregation and a single global threshold were insufficient for unsupervised federated botnet detection and motivated label-free, heterogeneity-aware aggregation with client-level threshold calibration.

ARTICLE INFO

Article history:

Received Sep 1, 2026
Revised Oct 5, 2026
Accepted Oct 12, 2026
Published Oct 31, 2026

Keywords:

Autoencoder
Botnet Detection
Federated Learning
Internet of Things
Non-IID Data

Copyright (c) 2026:

Muhammad Ikhsan,
Imam Riadi,
Ardiansyah Ardiansyah

License:

This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

*Corresponding Author
E-mail address: 2637083020@webmail.uad.ac.id

1. INTRODUCTION

The rapid expansion of the internet of things (IoT) has transformed services in smart homes, industry, healthcare, and transportation, yet most IoT devices have limited computation, memory, and energy budgets that leave little room for embedded security mechanisms [1-4]. These weaknesses are routinely exploited by botnets, which recruit large numbers of compromised devices to launch distributed denial of service (DDoS) attacks [5]. Six years of Mirai scanning activity showed that botnet behavior keeps evolving [6], and volumetric DDoS attacks increasingly affect the cloud and edge infrastructures on which IoT services depend [7,8].

Machine learning and deep learning based intrusion detection systems (IDS) have been widely proposed for IoT networks [4,9]. Most of them rely on a centralized architecture in which traffic from all devices is collected on a single server, which raises privacy concerns, consumes considerable

bandwidth, and creates a single point of failure [10,11]. Federated learning (FL) addresses these issues by training a shared model collaboratively while raw data remain on the clients and only model parameters are exchanged [12,13]. FL-based IDS have been applied to malware detection on N-BaIoT [14], zero-day botnet detection on IoT-edge devices [15], anomaly detection of IoT attacks [16], agricultural IoT [17], and decentralized DDoS detection [18], and recent reviews identify FL as a main direction for privacy-preserving collaborative intrusion detection [19,20].

The performance of FL degrades when client data are not independent and identically distributed (non-IID). Under non-IID data, local updates drift towards client-specific optima, which slows convergence and reduces the accuracy of the global model [21-24]. FedProx mitigates this drift with a proximal term [25], and controlled Dirichlet partitioning has become a standard way to emulate different degrees of heterogeneity [26]. In IoT networks, heterogeneity is not an artifact of simulation but an inherent property, because cameras, thermostats, doorbells, and industrial sensors generate distinct benign traffic profiles [27,28]. Several FL-IDS address non-IID data through client clustering, dynamic weighting, proximal regularization, or knowledge distillation [29,30], but these approaches are supervised and assume that labeled attack traffic is available on every client.

In practice, IoT traffic is rarely labeled. Label scarcity, together with heterogeneity and privacy constraints, has been identified as a key obstacle for federated intrusion detection [31]. Autoencoders trained only on benign traffic avoid this obstacle: they learn normal behavior and flag traffic with a large reconstruction error as anomalous, which also allows previously unseen attacks to be detected [32-34]. Unsupervised and semi-supervised FL-IDS have therefore begun to emerge, for example FedMSE, which combines a shrink autoencoder with a centroid one-class classifier [35], and G-PFL-ID, which personalizes a graph encoder for non-IID IoT systems [31]. Two practical issues, however, remain largely unquantified. First, the benign training data of a client may already contain botnet traffic when the device is compromised; such contamination is known to degrade unsupervised anomaly detectors [36] and resembles data poisoning in FL [37,38]. Second, the detection decision depends on a reconstruction-error threshold [39], and a single global threshold may not suit clients whose error distributions differ.

Existing evaluations of FL-IDS are mostly supervised, rely on one dataset, or consider a single heterogeneity setting. Table 1 positions this study against the closest FL-IDS studies. None of them evaluates an unsupervised federated detector under controlled levels of heterogeneity together with explicit training-data contamination and a comparison of global and client-level thresholds. As a result, the joint effect of these three factors has not been measured, and it is unclear whether new aggregation methods are needed and at which point of the pipeline they should intervene. The novelty of this study is therefore not a new detector but a controlled, label-free evaluation of these three factors and of their interaction. Its contributions are threefold:

- A reproducible benchmark protocol for unsupervised federated autoencoders on N-BaIoT and Edge-IIoTset that covers IID, natural per-device, and Dirichlet partitions combined with training-data contamination of 0% – 20% (264 training runs). Three design elements make the protocol suitable for unlabeled data: heterogeneity is created over unsupervised behavioral profiles rather than over class labels, contamination is drawn from an attack pool that is disjoint from the test pool, and all methods share the same client-level feature scaling.
- A quantitative analysis of the impact of heterogeneity and contamination on detection performance, inter-client fairness, training stability, and communication cost of FedAvg and FedProx relative to local-only and centralized training.
- An analysis of global versus client-level thresholds and of contaminated clients that identifies the failure modes of standard aggregation in unsupervised federated detection.

The remainder of this paper is organized as follows. Section 2 describes the datasets, partitioning, contamination, model, aggregation, and evaluation protocol. Section 3 presents and discusses the results, and Section 4 concludes the paper.

2. RESEARCH METHODS

This study followed a quantitative experimental design consisting of eight stages, as shown in Figure 1. Public datasets were preprocessed and partitioned into federated clients under several heterogeneity levels, contamination was injected into the benign training data of selected clients, and light-

weight autoencoders were trained locally and aggregated by standard FL algorithms. The resulting models were evaluated with global and client-level thresholds using detection, fairness, stability, and communication metrics.

Table 1. Positioning of this study against closely related FL-IDS studies.

Study	Learning setting	Heterogeneity	Training-data contamination	Threshold comparison	Datasets
Rey et al. [14]	Supervised MLP and unsupervised AE	Natural device split	Model poisoning by malicious clients	No	N-BaIoT
Popoola et al. [15]	Supervised DNN	Not addressed	No	No	Bot-IoT, N-BaIoT
Beshah et al. [29]	Supervised	Non-IID client clusters	No	No	IoTID20
Peng et al. [30]	Supervised	Non-IID	No	No	Edge-IIoTset, N-BaIoT
Nguyen and Beuran [35]	Semi-supervised AE on benign data	Non-IID	No	No	N-BaIoT
Oladele et al. [31]	Unsupervised graph encoder	Dirichlet and natural	No	No	IoT-23, N-BaIoT
This study	Unsupervised AE on benign data	IID, natural, Dirichlet α = 1.0, 0.5, 0.1	0% – 20% in 30% of clients	Global vs client-level	N-BaIoT, Edge-IIoTset

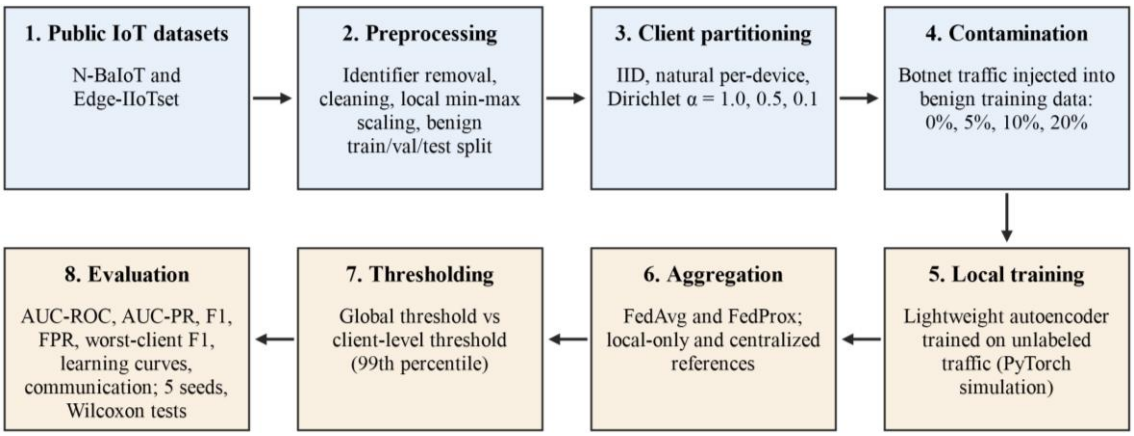


Figure 1. Research workflow of the benchmark.

2.1. Datasets

Two public datasets with complementary characteristics were used, as summarized in Table 2. N-BaIoT contains traffic of nine commercial IoT devices infected by the Mirai and BASHLITE botnets and describes each observation with 115 statistical traffic features [32]. Because benign and attack traffic are recorded per device, N-BaIoT provides a natural device-level partition and has been used in earlier FL-IDS studies [14,15]. Edge-IIoTset was designed for both centralized and federated learning and was collected from an IoT and IIoT testbed [40]. Its benign traffic is stored separately for ten sensors, which formed the natural partition, whereas its 14 attack types are not associated with individual sensors and were therefore treated as a shared attack pool.

2.2. Preprocessing

Records with missing or infinite values were removed from N-BaIoT. For Edge-IIoTset, identifier and payload fields such as timestamps, IP addresses, ports, checksums, raw acknowledgment numbers, URIs, DNS query names, and MQTT messages and topics were removed to prevent shortcut learning. Hexadecimal flags were converted to integers, the HTTP method was ordinally encoded, the remaining text fields were replaced by presence indicators, and missing values were set to zero, which

yielded 40 numeric features. Duplicate records were retained because flow-level features repeat frequently and de-duplication left only 33 – 219 benign records for some sensors; instead, at most 200,000 benign records per sensor and 100,000 records per attack type were randomly sampled. Benign traffic of each client was split into training, validation, and test subsets with a ratio of 70:10:20. Each client scaled its features with min–max normalization fitted only on its own training subset, so no raw statistics left the client. The same client-level scaling was applied to every method, so that differences between methods arose only from collaboration and aggregation.

Table 2. Characteristics of the datasets after preprocessing.

Dataset	Natural clients	Features	Benign / attack records	Attack coverage
N-BaIoT	9 devices	115	555,932 / 6,506,674	Mirai and BASHLITE (10 attack types), recorded per device
Edge-IIoTset	10 sensors	40	1,924,821 / 765,333	14 attack types in a shared pool

Attack records were divided into two disjoint halves: a test pool and a contamination pool. The test subset of each client contained its benign test records and the same number of attack records drawn from the test pool, using the attacks of the corresponding device in the natural N-BaIoT partition and the pooled attacks otherwise. Attack traffic was used in training only in the contamination scenarios described in Section 2.4.

2.3. Federated Partitioning

Three partitioning schemes were compared. In the IID scheme, benign samples were shuffled and distributed uniformly across $K = 10$ clients. In the natural scheme, each N-BaIoT device or Edge-IIoTset sensor formed one client. In the Dirichlet scheme, the benign traffic of each dataset was first grouped into $P = 10$ behavioral profiles using k-means clustering on log-compressed and standardized features. For each profile p, the proportions of its samples assigned to the $K = 10$ clients were sampled as in Equation (1):

$$q_p \sim \mathrm{Dir}(\alpha 1_K),\ p = 1, \ldots, P \tag{1}$$

where α is the concentration parameter. Smaller values of α concentrate each profile on fewer clients and therefore produce more heterogeneous clients [26]. The values α = 1.0, 0.5, and 0.1 were used to represent mild, moderate, and extreme heterogeneity, and partitions were redrawn until every client held at least 1,000 records.

2.4. Training-Data Contamination

Contamination emulated devices that were already compromised when their benign profile was learned. For 30% of the clients, i.e., three of ten clients chosen at random, attack records from the contamination pool were added to the benign training subset so that they formed a fraction ρ of the resulting training data, with ρ = 0%, 5%, 10%, and 20%. Because the contamination and test pools were disjoint, no injected record appeared in any test subset. Contamination was evaluated under the Dirichlet partition with α = 0.5. The injection is synthetic: attack records were drawn uniformly at random from the contamination pool, so every contaminated client received a mixture of all attack types at a fixed share. Real compromised devices may instead emit rarer, burstier, or single-family attack traffic whose share changes over time. The results therefore quantify the sensitivity to a controlled contamination level rather than reproduce a specific infection scenario.

2.5. Local Autoencoder

Each client trained a lightweight fully connected autoencoder with an encoder of d–64–32–16 neurons and a symmetric decoder, where d is the number of input features. ReLU activations were used in the hidden layers and a sigmoid activation in the output layer. The local objective of client k was the mean squared reconstruction error in Equation (2):

$$L_k(\theta) = \frac{1}{|D_k|}\sum_{x \in D_k}\left\|x - g_\theta\left(f_\theta(x)\right)\right\|_2^2 \quad (2)$$

where D_k is the training subset of client k, f_θ is the encoder, and g_θ is the decoder. The model contained 20,163 parameters for N-BaIoT (d = 115) and 10,488 parameters for Edge-IIoTset (d = 40), which corresponds to approximately 80.7 KB and 42.0 KB per transmitted update in 32-bit floating point. All methods used the Adam optimizer with a learning rate of 0.001, a batch size of 256, one local epoch per round, and R = 100 communication rounds.

2.6. Aggregation Methods

FedAvg aggregated the local models by weighting each client with its share of training samples, as in Equation (3) [12]:

$$w^{t+1} = \sum_{k=1}^{K}\frac{n_k}{n}w_k^{t+1} \quad (3)$$

where n_k is the number of training samples of client k and n is the total number of samples. FedProx added a proximal term to the local objective to limit client drift, as in Equation (4) [25]:

$$\min_w F_k(w) + \frac{\mu}{2}\|w - w^t\|^2 \quad (4)$$

where w^t denotes the global model received at round t. The value of μ was selected from 0.001, 0.01, and 0.1 in a preliminary run (seed 0, α = 0.5), in which μ = 0.001 gave the highest AUC-PR on both datasets; this value was used in all subsequent experiments. Two reference settings were included: local-only training, in which every client trained its own model for the same number of epochs without collaboration, and centralized training on the pooled training data of all clients, which ignores privacy constraints. The federated process was simulated in PyTorch on a single 16-core server, with all clients participating in every round. No physical gateways or network links were involved; communication was therefore computed from the model size (Section 2.8) rather than measured on a network.

2.7. Thresholding

The anomaly score of a sample x was its reconstruction error $s(x)$. Each client computed τ_k as the 99th percentile of the scores of its benign validation subset. The 99th percentile was chosen because it fixes the expected false positive rate on benign traffic at about 1% and can be computed from benign data alone, which matches the unsupervised setting; thresholds derived from the benign reconstruction-error distribution are standard practice for such detectors [39]. A higher percentile lowers false alarms at the cost of recall and vice versa, but the same percentile was applied to every client and method, so the choice does not favor either the global or the client-level threshold. In the global setting, the server combined these values into one threshold, as in Equation (5), whereas in the client-level setting each client used its own τ_k. A sample was classified as an attack when its score exceeded the threshold. For centralized training, the global threshold was computed from the pooled validation data, and for local-only training only client-level thresholds were meaningful.

$$\tau_G = \sum_{k=1}^{K}\frac{n_k}{n}\tau_k, \quad \tau_k = Q_{0.99}(\{s(x) : x \in V_k\}) \quad (5)$$

where V_k is the benign validation subset of client k and $Q_{0.99}$ denotes the 99th percentile.

2.8. Evaluation Metrics and Statistical Analysis

Detection performance was measured with precision, recall, F1-score, false positive rate (FPR), the area under the receiver operating characteristic curve (AUC-ROC), and the area under the precision–recall curve (AUC-PR). AUC-PR was computed on the test subset of each client and averaged over clients. F1-score and FPR were computed as in Equation (6):

$$\mathrm{F1} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \quad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \quad (6)$$

where TP, FP, TN, and FN are the numbers of true positives, false positives, true negatives, and false negatives. Inter-client fairness was measured by the worst-client F1-score and the standard deviation of F1-scores across clients. Training stability was assessed from learning curves of the pooled AUC-PR. The curves were evaluated every five rounds on a stratified subset of up to 5,000 benign and 5,000 attack test records per client, and the round of the peak value and the drop from the peak to the final round were reported. The communication cost was computed as $2 \cdot R \cdot K \cdot |w|$ parameters transmitted in total. Every configuration was repeated with five random seeds and reported as mean ± standard deviation. Because five seeds cannot reach significance in an exact two-sided Wilcoxon signed-rank test (minimum $p = 0.0625$), methods were compared with the exact Wilcoxon test over the ten dataset–partition settings, using seed-averaged values as paired observations. The effect of contamination was tested with the same test over ten seed-level pairs (two datasets × five seeds). Averaging the five seeds within each setting before testing prevented repeated runs from being treated as independent observations. Because several hypotheses were tested, *p*-values were adjusted with the Holm–Bonferroni procedure within each family of tests, namely seven method comparisons and six contamination comparisons, and a result was considered significant when the adjusted *p*-value was below 0.05. All significant results reported below remained significant after this adjustment, and also when all thirteen tests were treated as a single family (largest adjusted $p = 0.041$). Table 3 summarizes the experimental configuration.

Table 3. Experimental configuration.

Parameter	Value
Datasets	N-BaIoT (115 features), Edge-IIoTset (40 features)
Partitioning	IID; natural per device or sensor; Dirichlet α = 1.0, 0.5, 0.1 over P = 10 profiles
Number of clients	K = 10 (IID and Dirichlet); 9 (N-BaIoT natural); 10 (Edge-IIoTset natural)
Contamination	ρ = 0%, 5%, 10%, 20% in 30% of the clients (α = 0.5)
Local model	Autoencoder d–64–32–16–32–64–d, ReLU, sigmoid output
Training	Adam, learning rate 0.001, batch 256, 1 local epoch, 100 rounds
Methods	Local-only, FedAvg, FedProx (μ = 0.001), centralized
Threshold	99th percentile of benign validation scores; global and client-level
Repetitions	5 random seeds; exact Wilcoxon signed-rank tests with Holm–Bonferroni adjustment (significance level 0.05)
Total runs	264 (132 per dataset)

3. RESULTS AND DISCUSSIONS

This section first reports the effect of heterogeneity on detection performance and training stability, then the effect of contamination, followed by fairness and communication, and finally compares global and client-level thresholds.

3.1. Effect of Non-IID Heterogeneity

Table 4 presents the mean client AUC-PR of each method, and Figure 2 visualizes the same values. Collaboration through FedAvg did not outperform isolated training: local-only training achieved a higher AUC-PR than FedAvg in all ten dataset–partition settings, with a mean difference of 0.058 (Wilcoxon $p = 0.002$, Holm-adjusted $p = 0.014$), and centralized training was also higher than FedAvg in all ten settings (mean difference 0.033, adjusted $p = 0.014$). The largest gap occurred under

the natural N-BaIoT partition, where FedAvg reached 0.871 compared with 0.987 for local-only training. On Edge-IIoTset, the gap was largest at α = 0.1 (0.739 versus 0.882).

Table 4. Mean client AUC-PR (mean ± standard deviation over five seeds) under different partitioning schemes.

Dataset	Partition	Local-only	FedAvg	FedProx	Centralized
N-BaIoT	IID	0.981 ± 0.013	0.940 ± 0.004	0.999 ± 0.000	0.959 ± 0.010
N-BaIoT	α = 1.0	0.982 ± 0.007	0.941 ± 0.010	0.999 ± 0.000	0.959 ± 0.009
N-BaIoT	α = 0.5	0.990 ± 0.004	0.940 ± 0.014	0.999 ± 0.000	0.968 ± 0.011
N-BaIoT	α = 0.1	0.996 ± 0.003	0.958 ± 0.006	0.999 ± 0.000	0.987 ± 0.004
N-BaIoT	Natural	0.987 ± 0.002	0.871 ± 0.013	0.992 ± 0.001	0.964 ± 0.007
Edge-IIoTset	IID	0.722 ± 0.008	0.716 ± 0.028	0.726 ± 0.025	0.756 ± 0.045
Edge-IIoTset	α = 1.0	0.751 ± 0.017	0.709 ± 0.032	0.695 ± 0.008	0.715 ± 0.043
Edge-IIoTset	α = 0.5	0.770 ± 0.008	0.733 ± 0.020	0.701 ± 0.009	0.749 ± 0.056
Edge-IIoTset	α = 0.1	0.882 ± 0.030	0.739 ± 0.033	0.720 ± 0.028	0.800 ± 0.030
Edge-IIoTset	Natural	0.884 ± 0.009	0.821 ± 0.003	0.834 ± 0.002	0.839 ± 0.004

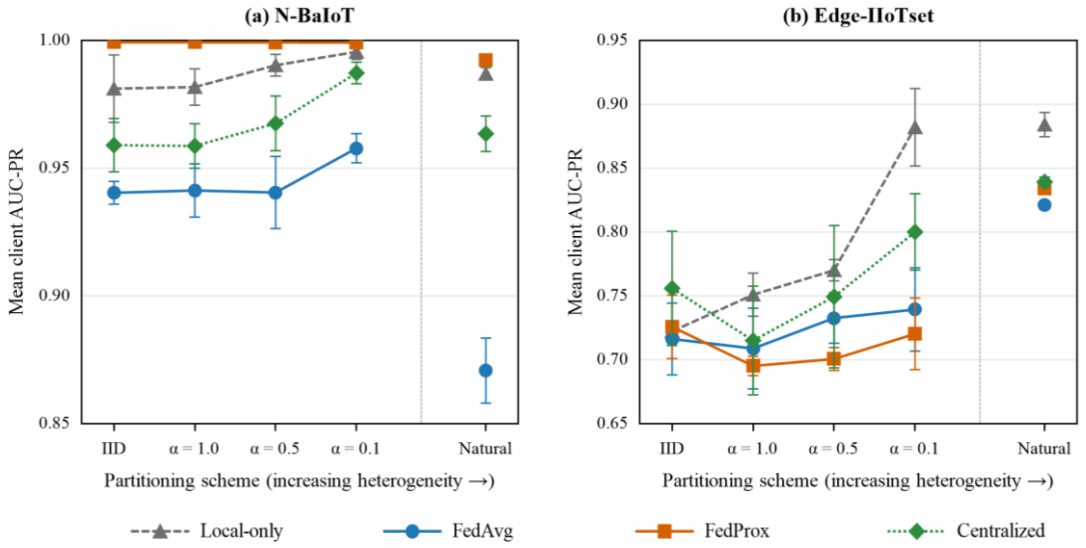


Figure 2. Mean client AUC-PR (mean ± standard deviation over five seeds) versus the partitioning scheme: (a) N-BaIoT and (b) Edge-IIoTset.

Contrary to expectation, the AUC-PR did not decrease monotonically as α decreased. For every method except FedProx on Edge-IIoTset (0.720 versus 0.726), the values at α = 0.1 were equal to or higher than those under IID data. Under extreme Dirichlet skew, each client holds only a few behavioral profiles, so its benign distribution is narrower and easier to separate from attack traffic when the evaluation is performed per client. In addition, the attack records of each test subset were drawn from the same pooled attack traffic for every client, so a client with one or two profiles faced a simpler normal-versus-attack separation than an IID client whose benign data mixed all ten profiles. The mean client AUC-PR therefore benefits from the narrower normal behavior created by extreme skew, whereas a single global threshold, which must serve clients with very different error levels, is penalized by it. Heterogeneity therefore did not primarily reduce the ranking ability of the models; its main effect appeared in the threshold, as shown in Section 3.4.

FedProx behaved differently on the two datasets. On N-BaIoT, it reached an AUC-PR of 0.992 – 0.999 in every partition and outperformed all other methods, whereas on Edge-IIoTset it was below FedAvg in three of five partitions. Across the ten settings, the difference between FedProx and FedAvg was not significant (mean difference 0.030, adjusted p = 0.262), and neither was the difference between local-only training and FedProx (adjusted p = 0.557). The learning curves in Figure 3 explain the N-BaIoT result. FedAvg reached its peak early, with a pooled AUC-PR of 0.96–1.00 be-

tween rounds 6 and 30, and then declined by 0.033 – 0.087 to its final value in every N-BaIoT partition. FedProx reached the same peak but did not decline (drop ≤ 0.001). Local-only and centralized training declined less (0.005 – 0.027). On Edge-IIoTset, all methods were stable, with drops of at most 0.047. The advantage of FedProx on N-BaIoT therefore reflected stability over many rounds rather than a better optimum. The client thresholds of all runs support this interpretation. On N-BaIoT, the 99th-percentile benign reconstruction error of the FedProx models was 1.8 – 5.4 times higher than that of the FedAvg models (geometric mean per partition), and the ratio exceeded 1 in all 25 seed–partition pairs. FedAvg therefore fitted the benign traffic much more tightly, which suggests that the shared model also learned to reconstruct attack traffic, a known weakness of reconstruction-based detectors [33]. On Edge-IIoTset, the ratio ranged from 0.6 to 2.1 without a consistent direction across seeds. This dataset has fewer features (40 versus 115) and several attack types whose reconstruction errors overlap with those of benign traffic (Section 3.3), so the proximal term had little separation to preserve. The benefit of FedProx thus depended on whether unconstrained aggregation tended to over-fit the benign data, which was the case for N-BaIoT but not for Edge-IIoTset.

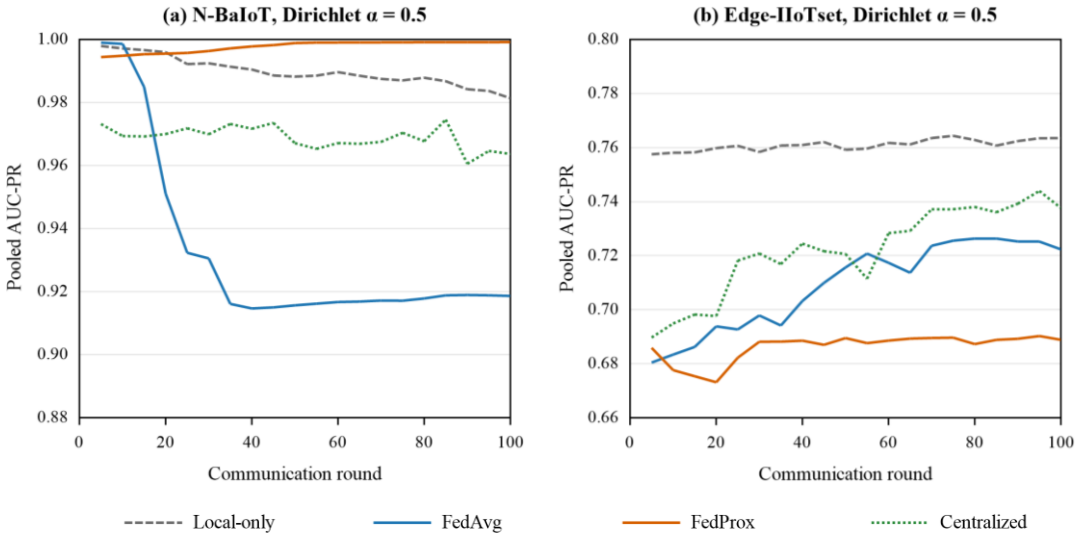


Figure 3. Learning curves of the pooled AUC-PR (mean over five seeds) under the Dirichlet partition with α = 0.5: (a) N-BaIoT and (b) Edge-IIoTset.

3.2. Effect of Training-Data Contamination

Table 5 reports the effect of increasing contamination at α = 0.5. Contamination degraded both FL methods. Comparing ρ = 20% with ρ = 0% over ten seed-level pairs, the mean client AUC-PR decreased by 0.053 for FedAvg and 0.064 for FedProx, and the worst-client F1-score with the global threshold decreased by 0.325 and 0.470, respectively (all adjusted p ≤ 0.012). FedProx was more sensitive than FedAvg on N-BaIoT: its worst-client F1-score fell from 0.983 to 0.137, below that of FedAvg (0.373), and its AUC-PR advantage over FedAvg shrank from 0.059 at ρ = 0% to 0.023 at ρ = 10% and was reversed at ρ = 20%. On Edge-IIoTset, FedProx kept a higher worst-client F1-score than FedAvg under contamination (0.341 – 0.356 versus 0.195 – 0.305).

The damage was concentrated on the contaminated clients but was not confined to them. In 59 of the 60 contaminated runs (two datasets, two methods, three contamination levels, and five seeds), the worst client was one of the contaminated clients with either threshold type, so the drop of the worst-client F1-score in Table 5 directly measures the harm done to these clients. For example, with FedProx on N-BaIoT at ρ = 20%, the contaminated clients reached a mean F1-score of 0.16, compared with 0.84 for the clean clients. With FedAvg on N-BaIoT, the mean recall of contaminated clients with the global threshold was 0.33 – 0.41, compared with 0.75 for clean clients. With FedProx at ρ = 20%, the recall of contaminated clients collapsed to 0.09. Clean clients were affected through aggregation: their mean AUC-PR under FedProx on N-BaIoT decreased from 0.999 at ρ = 0% to 0.910 at ρ = 20%. Contamination changed the FPR only slightly, by at most 2.1 percentage points, so it would not be revealed by the false alarm rate on benign data. Client-level thresholds did not repair this failure: with

FedAvg on N-BaIoT, the worst-client F1-score with client-level thresholds was only 0.03 – 0.18 for ρ ≥ 5%.

Table 5. Mean client AUC-PR, FPR with the global threshold (%), and worst-client F1-score with the global threshold under training-data contamination (α = 0.5; mean over five seeds).

Dataset	ρ (%)	AUC-PR FedAvg	AUC-PR FedProx	FPR FedAvg	FPR FedProx	Worst-client F1 FedAvg	Worst-client F1 FedProx
N-BaIoT	0	0.940	0.999	6.7	1.4	0.736	0.983
N-BaIoT	5	0.876	0.992	6.2	1.4	0.323	0.717
N-BaIoT	10	0.901	0.924	6.5	1.7	0.397	0.365
N-BaIoT	20	0.898	0.886	6.3	1.9	0.373	0.137
Edge-IIoTset	0	0.733	0.701	4.1	2.7	0.482	0.435
Edge-IIoTset	5	0.713	0.695	3.0	2.0	0.305	0.356
Edge-IIoTset	10	0.708	0.686	3.8	2.8	0.274	0.348
Edge-IIoTset	20	0.669	0.685	6.2	2.6	0.195	0.341

3.3. Fairness, Stability, and Communication

Table 6 summarizes inter-client fairness, training stability, and communication at α = 0.1. With the global threshold, the worst client of FedAvg reached an F1-score of 0.698 on N-BaIoT and 0.411 on Edge-IIoTset, 0.146 and 0.122 below the respective means. FedProx improved both the mean and worst-client F1-scores on N-BaIoT but not on Edge-IIoTset. The low F1-scores on Edge-IIoTset were consistent with its lower mean client AUC-ROC (0.57 – 0.84, not tabulated), which indicates that several of its attack types produce reconstruction errors similar to those of benign traffic. The peak round of FedAvg on N-BaIoT was 6 on average, after which the performance declined. Using an early round would therefore require a criterion for selecting the round without labels, which is not available in unsupervised FL. The communication cost of FedAvg and FedProx was identical by design, namely 161.3 MB per run on N-BaIoT and 83.9 MB on Edge-IIoTset. FedProx changes only the local objective, whereas the model size, the number of clients, and the number of rounds are the same, and the proximal term is computed on the client without being transmitted. Per client and round, this corresponds to sending and receiving one model of 80.7 KB on N-BaIoT or 42.0 KB on Edge-IIoTset, that is, about 161 KB or 84 KB of traffic in total, which is small for IoT gateways. The two methods therefore differ in computation and detection quality rather than in communication. However, the proximal term increased the training time by 34% – 42% (from 196 s to 278 s per run on N-BaIoT and from 637 s to 855 s on Edge-IIoTset at α = 0.5).

Table 6. Fairness (global threshold), stability, and communication at α = 0.1 (mean over five seeds).

Dataset	Method	Mean F1	Worst-client F1	Std. of F1	Peak round	Peak-to-final drop	Communication (MB)
N-BaIoT	FedAvg	0.844	0.698	0.095	6	0.033	161.3
N-BaIoT	FedProx	0.959	0.793	0.067	100	0.000	161.3
Edge-IIoTset	FedAvg	0.533	0.411	0.068	76	0.006	83.9
Edge-IIoTset	FedProx	0.514	0.385	0.082	61	0.006	83.9

3.4. Global Versus Client-Level Thresholds

Table 7 compares the global threshold in Equation (5) with client-level thresholds for FedAvg in all partitions. Under IID data, the two thresholds were almost equivalent. As heterogeneity increased, the FPR of the global threshold rose from 1.1% to 22.2% on N-BaIoT and from 1.5% to 18.1% on Edge-IIoTset at α = 0.1, and it reached 6.0% and 9.8% in the natural partitions. Client-level thresholds kept the FPR at or below 1.0% in every setting. The FPR reduction was significant for FedAvg (mean reduction of 6.7 percentage points, adjusted p = 0.014) and FedProx (4.8 percentage points, adjusted p = 0.016), and the F1-score did not decrease in nine of ten settings (mean increase

0.017, adjusted $p = 0.041$). The averaged global threshold was too low for clients whose benign reconstruction errors were higher than average, which produced most of the false alarms.

Table 7. Global versus client-level thresholds for FedAvg (mean ± standard deviation over five seeds; FPR in %).

Dataset	Partition	FPR global	FPR client-level	F1 global	F1 client-level
N-BaIoT	IID	1.1 ± 0.2	1.0 ± 0.1	0.834 ± 0.001	0.834 ± 0.001
N-BaIoT	α = 1.0	3.7 ± 2.8	1.0 ± 0.1	0.829 ± 0.003	0.833 ± 0.001
N-BaIoT	α = 0.5	6.7 ± 3.8	1.0 ± 0.1	0.828 ± 0.007	0.847 ± 0.020
N-BaIoT	α = 0.1	22.2 ± 8.2	1.0 ± 0.1	0.844 ± 0.025	0.895 ± 0.020
N-BaIoT	Natural	6.0 ± 0.9	1.0 ± 0.1	0.806 ± 0.023	0.793 ± 0.024
Edge-IIoTset	IID	1.5 ± 0.7	0.9 ± 0.1	0.408 ± 0.025	0.418 ± 0.042
Edge-IIoTset	α = 1.0	2.5 ± 0.5	0.9 ± 0.1	0.415 ± 0.029	0.431 ± 0.042
Edge-IIoTset	α = 0.5	4.1 ± 4.5	0.7 ± 0.2	0.523 ± 0.057	0.545 ± 0.048
Edge-IIoTset	α = 0.1	18.1 ± 12.1	0.6 ± 0.2	0.533 ± 0.045	0.566 ± 0.049
Edge-IIoTset	Natural	9.8 ± 1.5	0.2 ± 0.0	0.724 ± 0.024	0.754 ± 0.022

3.5. Discussion

Three observations emerge from these results. First, the degradation caused by heterogeneity was located mainly in the threshold rather than in the ranking of samples. AUC-PR did not decrease with α, whereas the FPR of the global threshold increased up to twentyfold. Calibrating thresholds per client is therefore a necessary, inexpensive component of unsupervised federated detection, and it requires sharing only one scalar per client. Second, contamination and legitimate heterogeneity produced different signatures. Heterogeneity raised the FPR under a global threshold, while contamination left the FPR almost unchanged but lowered the recall of the affected clients and leaked into clean clients through aggregation. A threshold cannot correct a model that has learned to reconstruct attacks, which suggests that contamination has to be addressed during aggregation rather than at the decision stage; such a mechanism was not evaluated in this study. Third, no fixed aggregation rule was best on both datasets. FedProx stabilized training on N-BaIoT but did not help on Edge-IIoTset and was more vulnerable to contamination, and FedAvg degraded silently after its early peak on N-BaIoT. Without labels, the server cannot observe either effect directly. These observations motivate, but do not test, aggregation rules that adapt to label-free signals such as the distribution of client reconstruction errors; developing and validating such rules is left to future work.

These results translate into practical guidance for deploying unsupervised federated detectors on IoT gateways. First, client-level thresholds should be the default. Each gateway stores a single scalar computed from its own benign validation data, which is inexpensive, but it introduces operational duties: the threshold must be recomputed when the benign behavior of a device drifts or its firmware changes, a newly joined client needs a period of benign traffic before its threshold is reliable, and a threshold computed on contaminated validation data would itself be too permissive. Second, local-only training should be evaluated before FL is adopted, because standard aggregation did not improve detection in any of the tested settings. Third, the global model should not simply be taken from the last round, since FedAvg degraded silently after an early peak on N-BaIoT. Fourth, a low false alarm rate is not evidence of a clean deployment, because contamination reduced recall without increasing the FPR.

These findings can be related to recent unsupervised FL-IDS. FedMSE weighted clients by the inverse mean squared error on a development set pooled from benign samples of all gateways [35], whereas G-PFL-ID relied on proximal and variance regularization with local personalization [31]. Neither approach was evaluated under explicit training-data contamination, which in this benchmark reduced the worst-client F1-score by up to 0.85. The results also show that local-only training is a strong baseline that should always be reported, because a federated method that does not exceed it gives no detection benefit to the participating clients.

This study has several limitations. The experiments were conducted in simulation on a single server rather than on physical gateways, so network latency, client dropout, and asynchronous updates

were not modeled, and both datasets were generated in laboratory testbeds. Contamination was injected synthetically as a fixed share of randomly sampled attack records. The measured degradation should therefore be read as the response to a controlled contamination level rather than as an estimate for a particular infection campaign, and evaluating traffic captured from actually compromised devices remains an important next step. The value of μ was selected with AUC-PR in a preliminary run because no labeled validation data exist in the unsupervised setting, which may favor FedProx slightly. Edge-IIoTset records were sampled with caps and retained duplicates, and its attacks formed a shared pool rather than device-specific traffic. CICIoT2023 [41], larger client populations, and additional partitioning schemes were not included. These limitations will be addressed in future work.

4. CONCLUSION

This study benchmarked, in simulation, unsupervised federated autoencoders for IoT botnet detection on N-BaIoT and Edge-IIoTset under IID, natural, and Dirichlet partitions combined with synthetically injected training-data contamination, using 264 runs with five seeds each. FedAvg did not outperform local-only training in any of the ten dataset–partition settings (mean AUC-PR difference of 0.058, Holm-adjusted $p = 0.014$). On N-BaIoT, it declined by up to 0.087 after reaching its peak within the first 30 rounds. FedProx with $\mu = 0.001$ prevented this decline and achieved an AUC-PR of up to 0.999 on N-BaIoT but gave no gain on Edge-IIoTset. Under heterogeneity, a single global threshold raised the FPR from about 1% to 22.2% and 18.1%, whereas client-level thresholds kept it at or below 1.0%. Contaminating 30% of the clients with 20% attack traffic reduced the worst-client F1-score by 0.33 for FedAvg and 0.47 for FedProx, and client-level thresholds did not recover this loss. These results indicate that standard aggregation and a single global threshold are insufficient for unsupervised federated botnet detection in heterogeneous IoT networks. Future work will develop a label-free, meta-adaptive aggregation method that distinguishes legitimate device heterogeneity from training-data contamination, stabilizes training without labeled validation data, and integrates federated, client-level threshold calibration.

AUTHOR CONTRIBUTIONS

Muhammad Ikhsan: Conceptualization, methodology, data curation, software implementation, investigation, formal analysis, visualization, and writing—original draft. Imam Riadi: Conceptualization, methodological development, supervision, validation, interpretation of findings, and writing—review and editing. Ardiansyah Ardiansyah: Methodology, experimental evaluation, validation, interpretation of results, and writing—review and editing. All authors contributed to manuscript revision, approved the final version, and agreed to be accountable for the accuracy and integrity of the work.

FUNDING STATEMENT

This research received no external funding.

DECLARATION OF AI USE

The authors declare that no AI tools were used in preparing this manuscript.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

All data supporting this study are included in the article. No separate datasets are available.

ETHICAL APPROVAL

Ethical approval was unnecessary, as no humans or animals were involved.

ACKNOWLEDGMENTS

The authors thank the Doctoral Program in Informatics, Universitas Ahmad Dahlan, for academic support, and Universitas Islam Negeri Sumatera Utara Medan for supporting the doctoral study of the first author. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for Internet of Things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, **23**(3), 1622-1658. DOI: https://doi.org/10.1109/comst.2021.3075439

[2] Khan, L. U., Saad, W., Han, Z., Hossain, E., & Hong, C. S. (2021). Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges. *IEEE Communications Surveys & Tutorials*, **23**(3), 1759-1799. DOI: https://doi.org/10.1109/comst.2021.3090430

[3] Kumari, P., & Jain, A. K. (2023). A comprehensive study of DDoS attacks over IoT network and their countermeasures. *Computers & Security*, **127**, 103096. DOI: https://doi.org/10.1016/j.cose.2023.103096

[4] Heidari, A., & Jabraeil Jamali, M. A. (2023). Internet of Things intrusion detection systems: a comprehensive review and future directions. *Cluster Computing*, **26**(6), 3753-3780. DOI: https://doi.org/10.1007/s10586-022-03776-z

[5] Gelgi, M., Guan, Y., Arunachala, S., Samba Siva Rao, M., & Dragoni, N. (2024). Systematic literature review of IoT botnet DDOS attacks and evaluation of detection techniques. *Sensors*, **24**(11), 3571. DOI: https://doi.org/10.3390/s24113571

[6] Affinito, A., Zinno, S., Stanco, G., Botta, A., & Ventre, G. (2023). The evolution of Mirai botnet scans over a six-year period. *Journal of Information Security and Applications*, **79**, 103629. DOI: https://doi.org/10.1016/j.jisa.2023.103629

[7] Yudhana, A., Riadi, I., & Suharti, S. (2022). Network forensics against volumetric-based distributed denial of service attacks on cloud and the edge computing. *International Journal of Safety and Security Engineering*, **12**(5), 577-588. DOI: https://doi.org/10.18280/ijsse.120505

[8] Wintolo, H., Riadi, I., & Yudhana, A. (2025). Enhancing private cloud security using knowledge understanding assessment defense method for distributed denial of service attack mitigation. *International Journal of Safety and Security Engineering*, **15**(11), 2323-2331. DOI: https://doi.org/10.18280/ijsse.151112

[9] Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, **4**(1), 18. DOI: https://doi.org/10.1186/s42400-021-00077-7

[10] Campos, E. M., Saura, P. F., González-Vidal, A., Hernández-Ramos, J. L., Bernabé, J. B., Baldini, G., & Skarmeta, A. (2022). Evaluating federated learning for intrusion detection in Internet of Things: Review and challenges. *Computer Networks*, **203**, 108661. DOI: https://doi.org/10.1016/j.comnet.2021.108661

[11] Agrawal, S., Sarkar, S., Aouedi, O., Yenduri, G., Piamrat, K., Alazab, M., Bhattacharya, S., Maddikunta, P. K. R., & Gadekallu, T. R. (2022). Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*, **195**, 346-361. DOI: https://doi.org/10.1016/j.comcom.2022.09.012

[12] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, **54**, 1273-1282. URL: https://proceedings.mlr.press/v54/mcmahan17a.html

[13] Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems*, **216**, 106775. DOI: https://doi.org/10.1016/j.knosys.2021.106775

[14] Rey, V., Sánchez Sánchez, P. M., Huertas Celdrán, A., & Bovet, G. (2022). Federated learning for malware detection in IoT devices. *Computer Networks*, **204**, 108693. DOI: https://doi.org/10.1016/j.comnet.2021.108693

[15] Popoola, S. I., Ande, R., Adebisi, B., Gui, G., Hammoudeh, M., & Jogunola, O. (2022). Federated deep learning for zero-day botnet attack detection in IoT-edge devices. *IEEE Internet of Things Journal*, **9**(5), 3930-3944. DOI: https://doi.org/10.1109/jiot.2021.3100755

[16] Mothukuri, V., Khare, P., Parizi, R. M., Pouriyeh, S., Dehghantanha, A., & Srivastava, G. (2022). Federated-learning-based anomaly detection for IoT security attacks. *IEEE Internet of Things Journal*, **9**(4), 2545-2554. DOI: https://doi.org/10.1109/jiot.2021.3077803

[17] Friha, O., Ferrag, M. A., Shu, L., Maglaras, L., Choo, K.-K. R., & Nafaa, M. (2022). FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things. *Journal of Parallel and Distributed Computing*, **165**, 17-31. DOI: https://doi.org/10.1016/j.jpdc.2022.03.003

[18] Alhasawi, Y., & Alghamdi, S. (2024). Federated learning for decentralized DDoS attack detection in IoT networks. *IEEE Access*, **12**, 42357-42368. DOI: https://doi.org/10.1109/access.2024.3378727

[19] Hernandez-Ramos, J. L., Karopoulos, G., Chatzoglou, E., Kouliaridis, V., Marmol, E., Gonzalez-Vidal, A., & Kambourakis, G. (2025). Intrusion detection based on federated learning: A systematic review. *ACM Computing Surveys*, **57**(12), 1-65. DOI: https://doi.org/10.1145/3731596

[20] Khraisat, A., Alazab, A., Singh, S., Jan, T., & Gomez, A., Jr. (2024). Survey on federated learning for intrusion detection system: Concept, architectures, aggregation strategies, challenges, and future directions. *ACM Computing Surveys*, **57**(1), 1-38. DOI: https://doi.org/10.1145/3687124

[21] Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. *Neurocomputing*, **465**, 371-390. DOI: https://doi.org/10.1016/j.neucom.2021.07.098

[22] Ma, X., Zhu, J., Lin, Z., Chen, S., & Qin, Y. (2022). A state-of-the-art survey on solving non-IID data in federated learning. *Future Generation Computer Systems*, **135**, 244-258. DOI: https://doi.org/10.1016/j.future.2022.05.003

[23] Lu, Z., Pan, H., Dai, Y., Si, X., & Zhang, Y. (2024). Federated learning with non-IID data: A survey. *IEEE Internet of Things Journal*, **11**(11), 19188-19209. DOI: https://doi.org/10.1109/jiot.2024.3376548

[24] Ye, M., Fang, X., Du, B., Yuen, P. C., & Tao, D. (2023). Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, **56**(3), 1-44. DOI: https://doi.org/10.1145/3625558

[25] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, **2**, 429-450. URL: https://proceedings.mlsys.org

[26] Li, Q., Diao, Y., Chen, Q., & He, B. (2022). Federated learning on non-IID data silos: An experimental study. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 965-978. DOI: https://doi.org/10.1109/icde53745.2022.00077

[27] Catillo, M., Pecchia, A., & Villano, U. (2023). A deep learning method for lightweight and cross-device IoT botnet detection. *Applied Sciences*, **13**(2), 837. DOI: https://doi.org/10.3390/app13020837

[28] Alsaleh, S. S., El Bachir Menai, M., & Al-Ahmadi, S. (2024). Federated learning-based model to lightweight IDSs for heterogeneous IoT networks: State-of-the-art, challenges, and future directions. *IEEE Access*, **12**, 134256-134272. DOI: https://doi.org/10.1109/access.2024.3460468

[29] Beshah, Y. K., Abebe, S. L., & Melaku, H. M. (2025). Dynamic weight clustered federated learning for IoT DDoS attack detection. *Scientific Reports*, **15**(1), 34036.

DOI: https://doi.org/10.1038/s41598-025-13204-y
[30] Peng, H., Wu, C., & Xiao, Y. (2025). FD-IDS: Federated learning with knowledge distillation for intrusion detection in non-IID IoT environments. *Sensors*, **25**(14), 4309. DOI: https://doi.org/10.3390/s25144309
[31] Oladele, D. A., Ige, A., Agbo-Ajala, O., Ekundayo, O., Thottempudi, S. G., Sibiya, M., & Mnkandla, E. (2026). G-PFL-ID: Graph-driven personalized federated learning for unsupervised intrusion detection in non-IID IoT systems. *IoT*, **7**(1), 13. DOI: https://doi.org/10.3390/iot7010013
[32] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, **17**(3), 12-22. DOI: https://doi.org/10.1109/mprv.2018.03367731
[33] Pang, G., Shen, C., Cao, L., & van den Hengel, A. (2021). Deep learning for anomaly detection: A review. *ACM Computing Surveys*, **54**(2), 1-38. DOI: https://doi.org/10.1145/3439950
[34] Lahasan, B., & Samma, H. (2022). Optimized deep autoencoder model for Internet of Things intruder detection. *IEEE Access*, **10**, 8434-8448. DOI: https://doi.org/10.1109/access.2022.3144208
[35] Nguyen, V. T., & Beuran, R. (2025). FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection. *Computers & Security*, **151**, 104337. DOI: https://doi.org/10.1016/j.cose.2025.104337
[36] Zhou, J., & Wu, Y. (2024). Outlier-probability-based feature adaptation for robust unsupervised anomaly detection on contaminated training data. *IEEE Transactions on Circuits and Systems for Video Technology*, **34**(10), 10023-10035. DOI: https://doi.org/10.1109/tcsvt.2024.3408034
[37] Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M. V., Herrera, F., & Martínez-Cámara, E. (2023). Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, **90**, 148-173. DOI: https://doi.org/10.1016/j.inffus.2022.09.011
[38] Thein, T. T., Shiraishi, Y., & Morii, M. (2024). Personalized federated learning-based intrusion detection system: Poisoning attack and defense. *Future Generation Computer Systems*, **153**, 182-192. DOI: https://doi.org/10.1016/j.future.2023.10.005
[39] Torabi, H., Mirtaheri, S. L., & Greco, S. (2023). Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, **6**(1), 1. DOI: https://doi.org/10.1186/s42400-022-00134-9
[40] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, **10**, 40281-40306. DOI: https://doi.org/10.1109/access.2022.3165809
[41] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, **23**(13), 5941. DOI: https://doi.org/10.3390/s23135941