Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
19/1 (2026), 107-120. DOI: http://dx.doi.org/10.21609/jiki.v19i1.1687

Federated Learning for Privacy-Preserving IoT Intrusion Detection under Extreme Non-IID Conditions

Michael Angello Qadosy Riyadi, Adinda Mariasti Dewi

Department of Information Technology, Directorate of Surabaya Campus, Telkom University (Surabaya Campus), Surabaya 60231, Indonesia

E-mail: michaelangello@student.telkomuniversity.ac.id, adindamariasti@student.telkomuniversity.ac.id

Abstract

The rapid growth of IoT devices has expanded attack surfaces, making intrusion detection critical. Traditional centralized IDS compromise privacy and strain bandwidth by requiring raw data transfer. Federated learning (FL) offers a privacy-by-design solution, enabling collaborative training across IoT clients while sharing only model updates. However, FL is highly sensitive to non-IID data. Extreme heterogeneity, prevalent in real-world IoT IDS datasets due to device-specific traffic patterns and severe class imbalances, causes significant convergence challenges and accuracy degradation. This study benchmarks four advanced FL algorithms (FedAvg, SCAFFOLD, FedYogi, and AdaFedAdam) on the RT-IoT2022 dataset (123,117 samples, 12 attack classes) under extreme non-IID conditions (Dirichlet $\alpha = 0.01$, average JSD = 0.5677, three heterogeneous clients). Using a multilayer neural network with 10-fold cross-validation nested in the FL loop, SCAFFOLD achieves the most stable performance (Round 100: accuracy 0.7981, F1-score 0.7451, ROC-AUC 0.9396), while FedAvg converges slowly (accuracy 0.6959). FedYogi and AdaFedAdam fail due to gradient starvation and second-moment explosion. Compared to centralized baselines (accuracy up to 1.000), FL incurs a 20% accuracy trade-off, an acceptable cost for enhanced privacy in edge-IoT environments. Contributions include the first validation of SCAFFOLD under extreme non-IID IoT IDS and a reproducible evaluation protocol.

Keywords: *federated learning, privacy-preserving, edge-IoT, Dirichlet, non-IID, neural networks*

1. Introduction

The rapid expansion of the Internet of Things (IoT) has transformed diverse sectors such as smart cities, Industry 4.0 and tele-health, yet the persistence of insecure and heterogeneous edge infrastructures has significantly enlarged the attack surface [1], [2], [3]. According to recent large-scale empirical research, backend systems utilising lightweight IoT protocols such as MQTT and CoAP exhibit considerable vulnerability: for example, only 0.16% of MQTT and XMPP back-ends employed TLS, and 30.38% of CoAP-speaking back-ends were susceptible to denial-of-service attacks [4], [5]. Likewise, systematic reviews report that IoT-based botnet-driven DDoS attacks increased more than five-fold within 12 months in some mobile networks, underscoring how connected "things" are increasingly leveraged for large-scale disruption [6], [7], [8]. Meanwhile, device-count forecasts indicate global endpoints may reach ~29 billion by 2030, stressing the urgent need for edge-centric intrusion detection and privacy-preserving security architectures [9].

Traditional machine learning-based intrusion detection systems (IDS) rely on centralized data aggregation at cloud servers for model training [10], [11]. This approach conflicts with stringent privacy regulations such as the EU General Data Protection Regulation (GDPR) 2016/679 and the California Consumer Privacy Act (CCPA) 2020, which mandate data processing at the source [12], [13], [14]. Moreover, bandwidth constraints in IoT networks—with average throughput of 100–500 kbps in LPWAN technologies, render raw data transfer technically infeasible [15], [16]. Empirical analysis by Celik *et al*. revealed that approximately 60% of IoT applications handle sensitive data flows, emphasizing the need for a new paradigm that can utilize distributed data while preserving privacy [17].

Federated Learning (FL), introduced by McMahan *et al*. in a seminal AISTATS paper, offers a paradigm shift by enabling collaborative model training without raw data exchange [18].

Only parameter updates (gradients or weights) are transmitted to a central server for aggregation [19], [20]. In this IoT IDS context, clients correspond to heterogeneous edge devices (e.g., smart sensors, gateways, or industrial controllers) that conduct local model training using their own generated network traffic data, while a central aggregator (typically an edge server or cloud coordinator) performs model aggregation without ever accessing raw data [21], [22], [23]. Although vanilla FL provides strong privacy-by-design benefits by eliminating the need for raw data transfer, privacy preservation is not inherent: parameter updates and aggregation remain susceptible to various attacks (e.g., inference attacks, model inversion, or gradient leakage), requiring the assumption of trusted (honest-but-curious) clients and aggregator, and often necessitating additional mechanisms such as differential privacy, secure multiparty computation, or homomorphic encryption in real-world deployments [24], [25]. In the IoT context, FL holds promise for privacy-preserving and scalable IDS. However, FL performance is highly sensitive to non-Independent and Identically Distributed (non-IID) data distributions [26], [27], [28], [29]. Extreme non-IID conditions are particularly prevalent in real-world IoT IDS datasets due to device-specific traffic patterns, varying operational environments, severe class imbalances across distributed edge nodes, and temporally dynamic attack behaviors, resulting in significant client drift, convergence difficulties, and substantial accuracy degradation [30], [31]. A study by Zhao *et al*. demonstrated accuracy drops of up to 40% under extreme non-IID conditions on datasets such as MNIST, CIFAR-10, and Keyword Spotting—none of which are IoT-specific [32]. Thus, realistic IoT datasets like RT-IoT2022, which exhibit severe class imbalance, require rigorous testing [33], [34].

Despite the observed performance degradation (approximately 20% accuracy drop compared to centralized baselines in extreme heterogeneous settings), the adoption of FL for IoT IDS is strongly justified beyond mere decentralization: it enables strict compliance with privacy regulations (e.g., GDPR and CCPA), drastically reduces bandwidth demands in resource-constrained networks, supports scalable training across billions of edge devices without raw data exposure, and facilitates practical deployment in environments where centralizing sensitive traffic data is either legally prohibited or technically impractical. This study aims to address these gaps by providing a rigorous comparative evaluation of state-of-the-art FL algorithms on a realistic IoT IDS dataset under extreme heterogeneity, alongside a reproducible methodological framework. The main contributions include: (1) the first empirical benchmark of advanced FL algorithms on extreme non-IID IoT IDS; (2) in-depth analysis of algorithm resilience and failure under client drift and gradient starvation; (3) development of an evaluation protocol with integrated cross-validation; and (4) design guidelines for FL deployment in edge computing systems. The findings are expected to lay the foundation for next-generation robust and widely adoptable FL-based IDS.

2. Related Work

2.1. Federated Learning Algorithms for Non-IID Data Distribution

Federated learning (FL) algorithms have evolved significantly to address the challenges of non-independent and identically distributed (non-IID) data, which are particularly acute in distributed IoT environments. The baseline FedAvg algorithm, while foundational, suffers from client drift and slow convergence under heterogeneous data distributions, prompting the development of specialized variants. For instance, FedProx introduces proximal regularization terms to enhance training stability and prevent divergence on clients with limited data [35], [36]. Similarly, SCAFFOLD employs control variates to estimate and correct the effects of client drift, demonstrating superior performance in scenarios with high statistical heterogeneity [37]. On the server side, adaptive optimizers such as FedYogi and FedAdam integrate momentum-based adjustments to gradient updates, aiming to mitigate variance in learning rates across rounds [38]. Empirical comparisons across diverse datasets reveal that SCAFFOLD consistently achieves faster convergence and higher accuracy than FedAvg in extreme non-IID settings, while FedYogi exhibits better stability than FedAdam when dealing with noisy gradients. However, these adaptive methods are not without limitations; they can encounter gradient starvation—where sparse client data leads to vanishing updates—or second-moment explosion, where accumulated variance causes instability, particularly in resource-constrained IoT nodes.

2.2. Federated Learning Applications in IoT Intrusion Detection Systems

The application of FL to intrusion detection systems (IDS) in IoT networks has gained momentum, driven by the need for privacy-preserving alternatives to centralized training.

Comprehensive surveys highlight FL's role in enabling collaborative threat detection across distributed edge devices without compromising sensitive traffic data [30], [39]. Most existing studies have benchmarked primarily vanilla FedAvg or its simple variants on IoT-specific datasets such as CICIoT2023, Edge-IIoTset, TON_IoT, and BoT-IoT, often achieving detection accuracies exceeding 95% under moderate heterogeneity conditions [40], [41]. However, systematic exploration of more advanced FL algorithms—designed specifically to handle severe statistical heterogeneity—remains scarce on realistic IoT IDS datasets. In particular, comparative evaluations under extreme non-IID settings (e.g., Dirichlet $\alpha \ll 0.1$), which better reflect device-specific traffic patterns and severe class imbalances in real-world deployments, are rarely reported. To bolster privacy beyond basic model aggregation, researchers have integrated differential privacy mechanisms to add calibrated noise to updates, secure multi-party computation for encrypted aggregation, and homomorphic encryption to enable computations on ciphertexts, effectively countering inference attacks like membership inference or model inversion [7], [8], [21], [40], [42]. Advanced architectures, such as clustered FL and personalized FL approaches (often combining FedProx with meta-learning), have been proposed to further mitigate non-IID effects and tailor models to individual device characteristics [43], [44]. Complementary techniques include blockchain-based secure aggregation and generative adversarial networks (GANs) for augmenting imbalanced attack classes, enhancing overall IDS robustness in dynamic IoT ecosystems [45].

2.3. Evaluation Methodologies and Challenges in FL-IDS Benchmarks

Evaluation protocols for FL-based IDS remain a critical area of development, with most studies relying on moderate heterogeneity levels simulated via Dirichlet distributions ($\alpha \geq 0.1$) or quantity-skew partitions on legacy datasets like NSL-KDD and CICIDS2017 [30]. While these approaches provide initial insights, they inadequately capture the extreme non-IID realities of modern IoT traffic, characterized by device-specific protocols (e.g., MQTT/CoAP variations) and severe class imbalances in datasets such as RT-IoT2022. Comparative assessments of multiple advanced algorithms—FedAvg, SCAFFOLD, FedYogi, and AdaFedAdam—under stringent conditions like Dirichlet $\alpha = 0.01$ are notably rare, and the integration of nested cross-validation within FL training loops, which is essential for unbiased generalization estimates, is rarely reported in the existing literature. Moreover, systematic analysis of failure modes, such as adaptive optimizer divergence due to sparse gradients in low-data clients, is underexplored, limiting the practical transferability of results to production edge deployments.

2.4. Identified Research Gaps and Contributions of the Present Work

Despite these advancements, several persistent gaps underscore the need for more rigorous FL-IDS research. Foremost is the scarcity of head-to-head benchmarks for state-of-the-art algorithms on contemporary, extreme non-IID IoT datasets like RT-IoT2022, which better reflect real-world traffic heterogeneity than surrogate benchmarks on CIFAR-10 or MNIST. Existing literature also falls short in dissecting adaptive optimizer pathologies—such as gradient starvation and second-moment instability—under authentic IoT constraints, and it rarely employs reproducible protocols with embedded cross-validation to ensure methodological robustness. Finally, while privacy enhancements are increasingly discussed, few studies quantify the privacy-utility trade-offs in IDS contexts or explore layered defenses beyond vanilla FL in scalable edge settings. The present work directly confronts these deficiencies by conducting the first comprehensive comparative evaluation of FedAvg, SCAFFOLD, FedYogi, and AdaFedAdam on the RT-IoT2022 dataset under extreme non-IID conditions (Dirichlet $\alpha = 0.01$), incorporating detailed failure mode analysis, a novel cross-validation-integrated evaluation framework, and actionable design guidelines for privacy-preserving FL-IDS deployment. This establishes a more reliable foundation for advancing robust, distributed threat detection in heterogeneous IoT networks.

3. Methodology

Figure 1 illustrates the research workflow, commencing with the data loading phase, followed by the encoding of categorical columns using LabelEncoder. Subsequently, the data is partitioned into three heterogeneous clients (non-IID) with random class distributions based on the Dirichlet distribution. The ensuing step involves 10-fold cross-validation, wherein for each fold, the data is split into training and testing sets, and both subsets are subsequently standardized using StandardScaler [46], [47]. Local models are then trained employing a multilayer neural network,

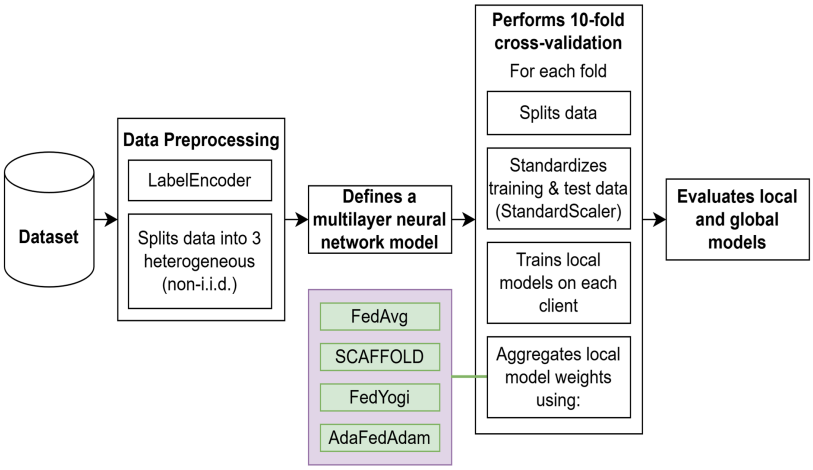

Figure 1. Research flow.

and the training outcomes from each client are aggregated using several federated learning algorithms, namely Federated Averaging (FedAvg), Stochastic Controlled Averaging for Federated Learning (SCAFFOLD), Federated Yogi (FedYogi), and Adaptive Federated Adam (AdaFedAdam). Finally, both local and global models are evaluated using various performance metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and log-loss.

3.1. Dataset

This study utilizes the RT-IoT2022 dataset, a public benchmark for intrusion detection in Internet of Things (IoT) environments that encompasses various realistic network attack types [48], [49]. The dataset comprises 123,117 instances with 83 features and 12 attack classes plus the normal class. The class distribution prior to client partitioning is presented in Table 1, which illustrates extreme class imbalance.

To simulate a federated learning scenario with extreme non-IID data, the dataset is partitioned into 3 heterogeneous clients using the Dirichlet distribution approach with concentration parameter α = 0.01 [50], [51]. The Dirichlet allocation proportions are described in Equation (1):

$$\boldsymbol{p_k} \sim \text{Dirichlet}(\boldsymbol{\alpha} \cdot \boldsymbol{1_K}), \quad \boldsymbol{\alpha = 0.01} \qquad (1)$$

where $\boldsymbol{p_k} = [\boldsymbol{p_{k1}}, \ldots, \boldsymbol{p_{kK}}]^{\top}$ is the proportion vector for client $\boldsymbol{k}$, and $\boldsymbol{K = 3}$ is the number of clients. The number of class $\boldsymbol{c}$ samples allocated to client $\boldsymbol{k}$ after the minimum stage is given in Equation (2):

$$\boldsymbol{n_{kc}} = \lfloor \boldsymbol{p_{kc}} \cdot (\boldsymbol{N_c} - \boldsymbol{K} \cdot \boldsymbol{m}) \rfloor, \quad \boldsymbol{m = 7} \qquad (2)$$

Table 1. Global class distribution in RT-IoT2022 dataset.

Class	Count	Proportion (%)
ARP_poisioning	7,750	6.29
DDOS_Slowloris	534	0.43
DOS_SYN_Hping	94,659	76.89
MQTT_Publish	4,146	3.37
Metasploit_Brute_Force_SSH	37	0.03
NMAP_FIN_SCAN	28	0.02
NMAP_OS_DETECTION	2,000	1.62
NMAP_TCP_scan	1,002	0.81
NMAP_UDP_SCAN	2,590	2.10
NMAP_XMAS_TREE_SCAN	2,010	1.63
Thing_Speak	8,108	6.59
Wipro_bulb	253	0.21

With $\boldsymbol{N_c}$ as the total samples of class $\boldsymbol{c}$ and adjustment on the last client to avoid remainder, the data partitioning yields three highly imbalanced clients with 104,160 (84.6%), 3,054 (2.5%), and 15,903 (12.9%) samples, respectively. The per-client class distribution is shown in Table 2. The Chi-Square test ($\boldsymbol{\chi^2 = 240827.93, df = 22, p = 0.00e + 00}$) indicates extreme heterogeneity (p < 1e-10), while Jensen–Shannon Divergence (JSD) values of 0.235486, 0.772346, and 0.695306 with an average of 0.567713 (>0.5) further confirm significant distributional differences across clients [52], [53]. These results demonstrate that the data partition is highly imbalanced and heterogeneous, reflecting strong non-IID conditions across clients.

3.2. Federated Learning (FL)

Federated Learning (FL) is a machine learning paradigm that enables distributed model training without requiring the transfer of raw data to a central server [54], [55]. This approach is desig-

Table 2. Class distribution across 3 heterogeneous clients.

Class	Client 1 (n=104,160)	Client 2 (n=3,054)	Client 3 (n=15,903)
ARP_poisioning	7 (0.01%)	7 (0.23%)	7,736 (48.64%)
DDOS_Slowloris	519 (0.50%)	7 (0.23%)	8 (0.05%)
DOS_SYN_Hping	94,644 (90.86%)	7 (0.23%)	8 (0.05%)
MQTT_Publish	4,131 (3.97%)	7 (0.23%)	8 (0.05%)
Metasploit_Brute_Force_SSH	21 (0.02%)	7 (0.23%)	9 (0.06%)
NMAP_FIN_SCAN	7 (0.01%)	7 (0.23%)	14 (0.09%)
NMAP_OS_DETECTION	7 (0.01%)	1,985 (65.00%)	8 (0.05%)
NMAP_TCP_scan	7 (0.01%)	987 (32.32%)	8 (0.05%)
NMAP_UDP_SCAN	2,576 (2.47%)	7 (0.23%)	7 (0.04%)
NMAP_XMAS_TREE_SCAN	1,995 (1.92%)	7 (0.23%)	8 (0.05%)
Thing_Speak	7 (0.01%)	19 (0.62%)	8,082 (50.82%)
Wipro_bulb	239 (0.23%)	7 (0.23%)	7 (0.04%)

ned to preserve data privacy by training local models on devices or clients, subsequently sending only parameter updates (weights) to the server for aggregation into a global model. In addition to its privacy-preserving advantages, FL has the potential to yield a global model with superior performance compared to models trained locally on individual clients. This is achieved through the aggregation of knowledge from heterogeneously distributed data, enabling the global model to capture more general and robust patterns than local models, which learn only from limited data subsets on each client.

In contrast to centralized learning, where all data is collected and stored on a single central server for training, centralized systems require data upload to the server, posing privacy and security risks due to potential leakage of sensitive information [56], [57]. Figure 2 illustrates the weaknesses of the centralized approach. Moreover, Federated Learning minimizes bandwidth usage by sharing only model parameters and supports continuous adaptation in dynamic, heterogeneous environments.

The FL process comprises several key steps, including local training on each client using its respective data, transmission of local model weights to the server, weight aggregation to update the global model, and redistribution of the global model to clients for subsequent iterations until convergence. To clarify this workflow, Figure 3 depicts the general Federated Learning process.

In alignment with the standard federated learning framework, this study assumes an honest-but-curious (semi-honest) threat model for both the participating clients and the central aggregator [58], [59]. Under this assumption, all entities faithfully execute the prescribed protocol and perform computations as intended, without deviating from or disrupting the training process [60], [61]. However, they may attempt to infer sensitive information from the data or model updates they legitimately receive, such as inferring private patterns from aggregated gradients or the global model parameters. This threat model is widely adopted in federated learning literature as it balances practical deployability with privacy considerations, while allowing the focus to remain on addressing core challenges such as extreme non-IID data distributions and resource heterogeneity in IoT environments. Malicious behaviors are out of scope for this work.

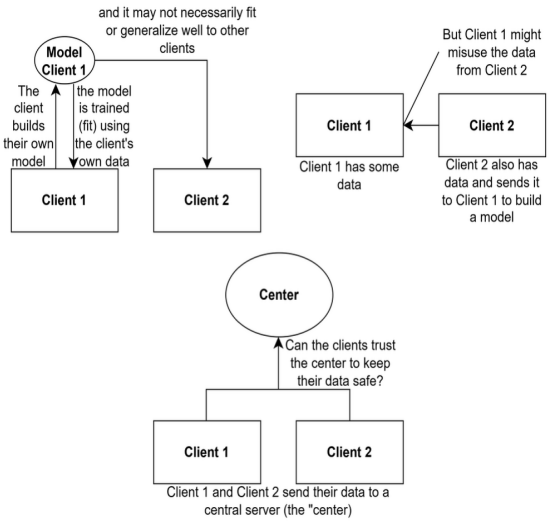

Figure 2. Weakness of centralized learning.

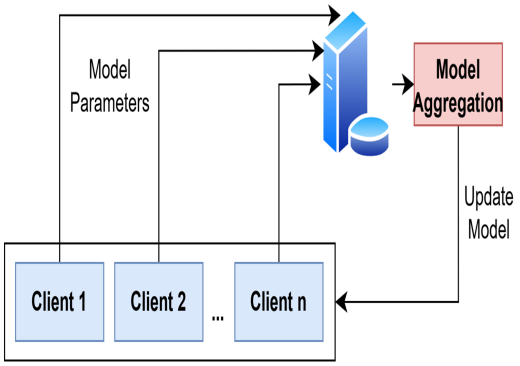

Figure 3. General federated learning workflow.

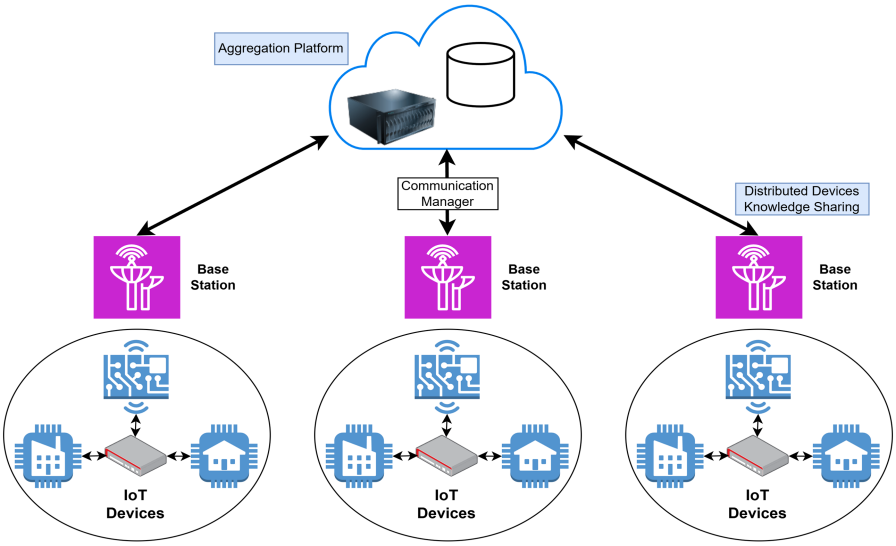

Figure 4. Architecture of federated learning-based IDS in edge-IoT environments.

In the context of IoT intrusion detection systems (IDS), the federated learning architecture is adapted to accommodate heterogeneous edge devices operating in resource-constrained and privacy-sensitive environments. As illustrated in Figure 4, clients consist of diverse IoT endpoints—such as smart sensors, gateways, industrial controllers, or connected appliances—that capture and process local network traffic data for intrusion detection. Each client trains a local model independently on its device-generated data, transmitting only encrypted model updates (gradients or weights) to a central aggregator (typically an edge server or cloud coordinator). The aggregator performs model fusion without accessing raw traffic flows, thereby preserving data locality and mitigating privacy risks associated with centralized data transfer.

It is important to note that, in practical IoT deployments, not all IoT devices are capable of performing local model training required by federated learning. In this study, the term "client" refers to edge-capable IoT nodes or IoT gateways equipped with sufficient computational resources to support on-device training, rather than highly resource-constrained sensing devices. Such clients typically include single-board computers or embedded edge platforms, such as Raspberry Pi 4, NVIDIA Jetson Nano, or functionally equivalent edge devices, which have been widely adopted in federated learning–based IoT applications due to their multi-core CPUs, adequate memory capacity, and, in some cases, GPU acceleration [62], [63], [64], [65]. Lightweight IoT sensors and microcontroller-based devices are assumed to operate primarily as data acquisition or inference nodes, forwarding collected traffic features to nearby edge-capable clients for local training. This assumption reflects realistic IoT system architectures and ensures the feasibility of executing federated learning under resource constraints while preserving data locality and privacy.

This study implements four Federated Learning algorithms to aggregate local model weights into a global model, namely Federated Averaging (FedAvg), Stochastic Controlled Averaging for Federated Learning (SCAFFOLD), Federated Yogi (FedYogi), and Adaptive Federated Adam (AdaFedAdam).

3.3. Federated Averaging (FedAvg)

Federated Averaging (FedAvg) is an FL algorithm that aggregates local model weights from each client using a weighted average, with weights proportional to the amount of data possessed by each client [66]. The weight aggregation process in FedAvg is described in Equation (3):

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

where w^{t+1} is the global model weights at iteration $t+1$, w_k^t is the local model weights from client k at iteration t, n_k is the number of data points on client k, and $n = \sum_{k=1}^{K} n_k$ is the total number of data points across all clients.

3.4. Stochastic Controlled Averaging for Federated Learning (SCAFFOLD)

SCAFFOLD is an FL algorithm designed to improve convergence in heterogeneous data settings by introducing control variables to reduce

gradient variance across clients [67]. This algorithm employs local c_k and global c control variables to correct the local model update direction, thereby accelerating convergence compared to FedAvg in non-IID scenarios.

The weight aggregation process in SCAFFOLD is described in Equation (4):

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t \quad (4)$$

Equation (4) shows that weight aggregation in SCAFFOLD is similar to FedAvg, using a weighted average of local model weights. However, SCAFFOLD introduces a correction mechanism via control variables to adjust local model updates. The update of the local control variable is computed using Equation (5):

$$c_k^{t+1} = c_k^t - c^t + \frac{1}{\eta}(w^t - w_k^{t+1}) \quad (5)$$

where c_k^t is the control variable of client k at iteration t, c^t is the global control variable, η is the learning rate, w^t is the global model weights at iteration t, and w_k^{t+1} is the local model weights of client k after local training at iteration $t+1$.

3.5. Federated Yogi (FedYogi)

Federated Yogi (FedYogi) is an adaptive variant of FedAvg that integrates the Yogi optimizer to better handle data heterogeneity [68]. FedYogi adjusts the global model weight updates based on gradient magnitude, incorporating a regularization parameter to prevent unstable updates. The weight update in FedYogi is described in Equation (6):

$$w^{t+1} = w^t + \eta \frac{\Delta^t}{\sqrt{\sum(\Delta^t)^2} + \tau} \quad (6)$$

where w^{t+1} is the global model weights at iteration $t+1$, w^t is the global model weights at iteration t, $\Delta^t = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t - w^t$ is the average gradient from all clients and τ is the regularization parameter to prevent division by zero.

3.6. Adaptive Federated Adam (AdaFedAdam)

Adaptive Federated Adam (AdaFedAdam) integrates the Adam optimizer into the FL framework to enhance efficiency and convergence [38]. This algorithm uses the first moment (moving average of gradients) and the second moment (moving average of squared gradients) to adaptively adjust weight updates.

The weight update process in AdaFedAdam involves three steps: updating the first moment, updating the second moment, and updating the weights. These steps are described in Equations (7), (8), and (9):

$$m^{t+1} = \beta_1 m^t + (1 - \beta_1)\Delta^t \quad (7)$$

$$v^{t+1} = \beta_2 v^t + (1 - \beta_2)(\Delta^t)^2 \quad (8)$$

$$w^{t+1} = w^t + \eta \frac{m^{t+1}/(1-\beta_1^{t+1})}{\sqrt{v^{t+1}/(1-\beta_2^{t+1})} + \epsilon} \quad (9)$$

where m^{t+1} is the first moment at iteration $t+1$, v^{t+1} is the second moment, $\Delta^t = \sum_{k=1}^{K} \frac{n_k}{n} w_k^t - w^t$ is the average gradient from all clients, β_1 and β_2 are decay parameters for the first and second moments and ϵ is a small constant to prevent division by zero. Equation (7) computes the first moment as a linear combination of the previous moment and the new gradient. Equation (8) computes the second moment to capture gradient variance. Equation (9) updates the global model weights by adjusting the update step based on bias-corrected first and second moments.

3.7. Local Model Architecture: Neural Network

Table 3 summarizes the layers in the local model architecture, including the number of parameters and output shapes. The local model employed within the Federated Learning (FL) framework in this study is a multilayer neural network [69]. This neural network is composed of four main layers: three hidden layers with 128, 64, and 32 neurons, respectively, and a final output layer with 12 neurons corresponding to the number of classes in the classification task [70], [71].

Table 3. Local neural network architecture.

Layer	Layer Type	Output Shape	Number of Parameters
Dense (ReLU)	Dense Layer (ReLU Activation)	(None, 128)	10,752
Dropout	Dropout Layer	(None, 128)	0
Dense (ReLU)	Dense Layer (ReLU Activation)	(None, 64)	8,256
Dropout	Dropout Layer	(None, 64)	0
Dense (ReLU)	Dense Layer (ReLU Activation)	(None, 32)	2,080
Dense (Softmax)	Output Layer (Softmax Activation)	(None, 12)	396

The use of a neural network in this study is motivated by its parameterized structure, which

facilitates efficient aggregation in the Federated Learning framework. Neural networks represent knowledge through adjustable weights and biases, making them well-suited for the parameter-sharing mechanism of FL [72], [73].

3.8. Experimental Setup

To provide a structured and clear overview, the experimental procedure is summarized in Figure 5. This algorithm encompasses all steps from data loading to evaluation.

Input: Dataset (D), Number of clients ($K = 3$), Number of cross-validation folds ($F = 10$), FL algorithm ($\in$ {FedAvg,SCAFFOLD,FedYogi,AdaFedAdam})
Output: Global model (w^{global}), Evaluation metrics: $\{accuracy, precision, recall, F1-score, ROC-AUC, log-loss\}$
1. Data Loading and Model Initialization: - Load dataset (D) - Partition dataset (D) into ($K = 3$) heterogeneous (non-IID) subsets using Dirichlet distribution - Initialize global model (w^{global}) - Distribute (w^{global}) to each client (k) 2. Training with 10-Fold Cross-Validation and Evaluation: - For each fold ($f \in \{1, 2, \dots, F\}$): - Split local data of each client (k) into training (D_k^{train}) and testing (D_k^{test}) - For each client ($k \in \{1, 2, 3\}$): - Train local model (w_k) on (D_k^{train}) - Send local model weights (w_k) to the server - Aggregate local model weights at the server to update (w^{global}) - Distribute updated (w^{global}) to each client (k) - For each fold ($f \in \{1, 2, \dots, F\}$): - Evaluate local model for each client (k) - Evaluate global model - Compute performance metrics - Aggregate performance metrics as the average across all folds.

Figure 5. Federated learning with 10-fold cross-validation on Non-IID RT-IoT2022 dataset.

To evaluate the performance of local and global models within the proposed Federated Learning (FL) framework, this study employs several standard performance metrics: accuracy, precision, recall, F1-score, ROC-AUC, and log-loss [74], [75], [76]. These metrics provide a comprehensive assessment of both classification effectiveness and model reliability.

Evaluation begins with accuracy, which measures the proportion of correct predictions out of total predictions made by the model. Accuracy is computed using Equation (10):

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (10)$$

This equation computes accuracy as the ratio of correctly predicted samples—true positives (TP, positive samples correctly predicted as positive) plus true negatives (TN, negative samples correctly predicted as negative)—to the total number of samples, which includes TP, TN, false positives (FP, negative samples incorrectly predicted as positive), and false negatives (FN, positive samples incorrectly predicted as negative).

Next, precision is computed to evaluate the model's exactness in identifying positive classes. Precision is calculated using Equation (11):

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (11)$$

Precision reflects the proportion of positive predictions that are actually positive.

To complement precision, recall is computed to assess the model's ability to detect all positive samples. Recall is calculated using Equation (12):

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (12)$$

Recall indicates how well the model identifies all positive cases.

To balance precision and recall, the F1-score is used as the harmonic mean of both metrics, providing a more robust performance measure under imbalanced class distributions. The F1-score is computed using Equation (13):

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \quad (13)$$

This equation giving equal weight to both metrics. ROC-AUC is computed based on the Receiver Operating Characteristic (ROC) curve, which plots the relationship between the true positive rate (equivalent to recall) and the false positive rate (FPR). FPR is computed using Equation (14):

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \quad (14)$$

ROC-AUC is calculated as the area under the ROC curve, with values close to 1 indicating excellent discriminative ability.

Finally, log-loss is used to measure the uncertainty of the model's predictions based on the output class probabilities. Log-loss is computed using Equation (15):

$$\text{Log-loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j} \log\left(p_{i,j}\right) \quad (15)$$

where N is the number of samples, C is the number of classes, $y_{i,j}$ is a binary indicator (1 if sample i belongs to class j, 0 otherwise), and $p_{i,j}$ is the model's predicted probability for sample i

in class j.

4. Results and Analysis

This chapter presents an in-depth empirical evaluation of four federated learning algorithms—FedAvg, SCAFFOLD, FedYogi, and AdaFedAdam—in comparison with local models using the RT-IoT2022 dataset under extreme non-IID partitioning (Dirichlet $\alpha = 0.01$, three clients). The evaluation focuses on convergence behavior, robustness to severe data heterogeneity (average JSD of 0.5677), and both theoretical and practical implications for Internet of Things (IoT) intrusion detection systems (IDS). Experimental results highlight distinct stability and optimization characteristics among the federated methods under constrained client distributions. All performance metrics are averaged over 10-fold cross-validation at selected communication rounds to ensure statistical robustness.

Table 4 presents model performance at Rounds 1, 50, and 100, while longitudinal trends are illustrated in Figure 6 to further clarify convergence and divergence patterns over training iterations under the proposed framework. This comparison allows for a clearer identification of performance trade-offs across different stages of the training process.

Table 4. Average performance of local and global models at rounds 1, 50, and 100 (averaged across 10 folds).

Round	Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	log loss
1	Local	0.3301	0.2988	0.3301	0.2849	0.6641	2.6621
	FedAvg	0.4030	0.5700	0.4030	0.4060	0.7453	2.2577
	SCAFFOLD	0.8040	0.7409	0.8040	0.7571	0.9440	0.5838
	FedYogi	0.1214	0.2112	0.1214	0.1332	0.4437	2.5790
	AdaFedAdam	0.0609	0.0672	0.0609	0.0571	0.4425	2.5941
50	Local	0.3364	0.3194	0.3364	0.2894	0.6552	2.6228
	FedAvg	0.5660	0.6877	0.5660	0.5845	0.7082	2.2201
	SCAFFOLD	0.7946	0.7206	0.7946	0.7372	0.9450	0.5774
	FedYogi	0.0657	0.3685	0.0657	0.0893	0.4762	14.7965
	AdaFedAdam	0.2729	0.2019	0.2729	0.2193	0.5329	2.4659
100	Local	0.3432	0.3449	0.3432	0.3021	0.6503	2.5871
	FedAvg	0.6959	0.6933	0.6959	0.6643	0.7086	2.1772
	SCAFFOLD	**0.7981**	**0.7345**	**0.7981**	**0.7451**	**0.9396**	**0.5882**
	FedYogi	0.0639	0.2838	0.0639	0.0841	0.4901	14.9096
	AdaFedAdam	0.2069	0.1295	0.2069	0.1465	0.4770	2.4656

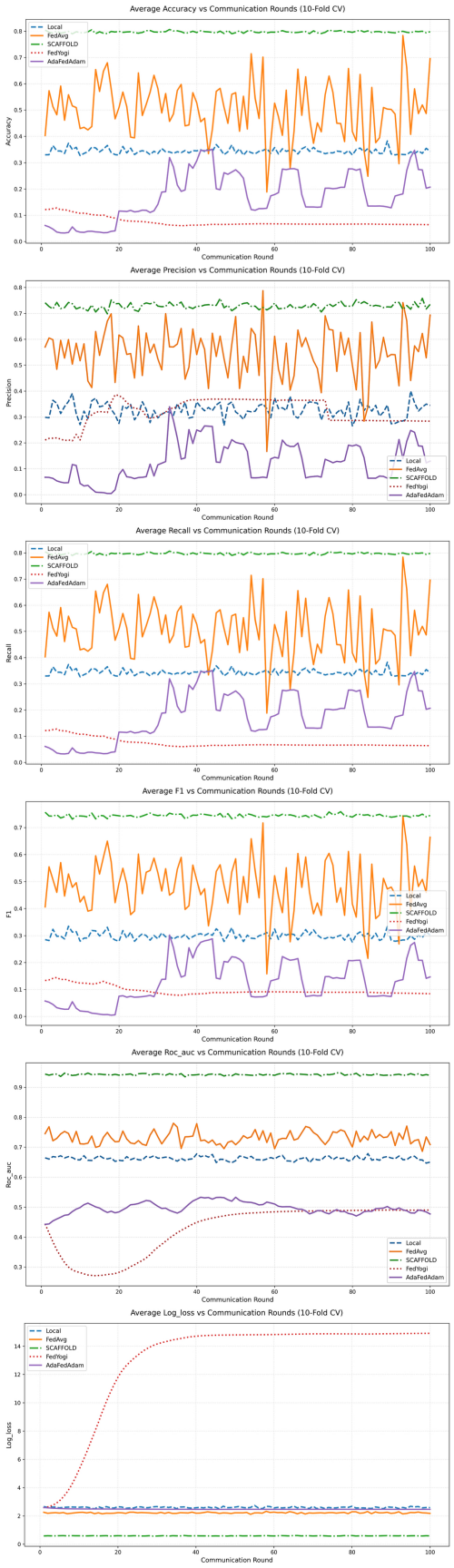

Figure 6. Convergence trends of key metrics in federated learning (rounds 1–100).

Analysis of Table 4 and Figure 6 reveals consistently high performance of SCAFFOLD throughout the training cycle. The algorithm maintains stable metric values from early rounds, with control variables effectively mitigating client drift caused by the dominance of Client 1 (84.6% of data, 90.86% DOS_SYN_Hping) and extreme scarcity in Client 2 (only 7 samples per minority class). This stability is reflected in minimal variance across discriminative (ROC-AUC $\geq$ 0.9396) and probabilistic (Log-loss $\leq$ 0.5882) metrics, positioning SCAFFOLD as a reliable choice for federated learning on extreme non-IID data in the IoT IDS domain.

FedAvg exhibits slow but consistent convergence, with significant improvement after Round 50. This behavior reflects the ability of weighted aggregation to gradually incorporate rare signals from minority clients, albeit at the cost of intensive communication. At Round 100, FedAvg achieves lower performance than SCAFFOLD but demonstrates substantial gains over local models, affirming its practicality in bandwidth-constrained systems where convergence latency is tolerable.

In contrast, FedYogi and AdaFedAdam experience systematic failure. FedYogi displays explosive divergence accelerated by error accumulation in second-moment estimation (β_2), particularly on sporadic gradients from rare classes. This aligns with the hypothesis that dual-momentum adaptive optimizers are vulnerable to gradient starvation in small clients. AdaFedAdam, while more stable, becomes trapped in suboptimal plateaus due to failure in normalizing global updates against inter-client variance, resulting in poor generalization on minority classes.

Local models, serving as the baseline, fail to surpass basic generalization thresholds, remaining confined to local distribution bias (e.g., over-reliance on DOS_SYN_Hping). This underscores the intrinsic value of federated learning in enabling cross-client knowledge transfer without compromising privacy, particularly in edge-computing IoT architectures.

For context against centralized approaches, Table 5 summarizes recent studies on the RT-IoT2022 dataset, highlighting the typical performance of fully centralized training paradigms. These results provide a useful reference point under centralized data assumptions and reflect performance under idealized training conditions. As such, they are included primarily to contextualize the comparison with distributed and federated learning settings.

Table 5. Performance comparison with centralized state-of-the-art IDS on RT-IoT2022 dataset.

Author	Year	Objective	Results
[77]	2023	Develop lightweight IDS for IoT using quantized autoencoders (QAE-u8, QAE-f16) and Zeek flow preprocessing; centralized training on RT-IoT2022.	Acc: 0.9840, Prec: 0.9839, Rec: 0.9840, F1: 0.9839
[31]	2024	Improve IDS via feature-selection pipeline and ML classifiers trained centrally on RT-IoT2022.	Acc: 0.9640, Prec: 0.9740, Rec: 0.8710, F1: 0.9190
[78]	2024	Assess multiple ML models (SVM, KNN, NB, DT) statistically using skewness, kurtosis, PCC, IGR on RT-IoT2022.	Acc: 0.9800 (SVM), 0.9974 (KNN), 0.8250 (NB), 0.9985 (DT)
[79]	2025	Apply semi-supervised ML with entropy filtering; tested on multiple datasets incl. RT-IoT2022.	Acc: 1.000 (DT)
[80]	2025	Enhance dataset balance using cosine similarity and hybrid DL (RegNet) models on RT-IoT2022.	Acc: 0.9836, Prec: 0.9900, Rec: 0.9776, F1: 0.9838
This study	2025	Conduct the first comparative benchmark of four advanced FL algorithms (FedAvg, SCAFFOLD, FedYogi, AdaFedAdam) on RT-IoT2022 with extreme non-IID partitioning.	Global model (Round 100, 10-fold avg): SCAFFOLD Acc: 0.7981, Prec: 0.7345, Rec: 0.7981, F1: 0.7451, ROC-AUC: 0.9396, Log-loss: 0.5882; FedAvg Acc: 0.6959, F1: 0.6643; FedYogi/AdaFedAdam diverge or plateau <0.21 Acc

Analysis of Table 5 reveals a clear performance trade-off in distributed privacy-preserving settings. Centralized approaches benefit from unrestricted access to aggregated data, enabling superior feature engineering, model optimization, and handling of class imbalances, resulting in near-perfect detection rates (up to 1.000 accuracy). In contrast, the federated paradigms evaluated here operate under severe constraints—no raw data exchange, extreme non-IID partitioning, and limited communication—yet achieve competitive results, with SCAFFOLD reaching 0.7981 accuracy and 0.9396 ROC-AUC. This ~20% accuracy gap, while notable, is an expected consequence of prioritizing data locality and basic privacy-by-design (only model updates transmitted). As discussed earlier, vanilla FL mitigates raw data exposure but remains vulnerable to advanced inference attacks on

gradients; thus, the privacy gains are substantial yet not absolute without additional layered defenses. SCAFFOLD's strong discriminative power (high ROC-AUC) suggests practical viability for edge-IoT deployment, where preventing data breaches and ensuring regulatory compliance (e.g., GDPR) often outweigh marginal accuracy improvements.

The primary contributions of this study include the first empirical validation of SCAFFOLD on an IoT IDS dataset under extreme heterogeneity ($\alpha = 0.01$, JSD > 0.5), achieving robust performance without raw data access. It further quantifies FedAvg convergence bounds as a function of communication rounds, offering design guidance for resource-constrained systems. The identification of adaptive optimizer failure mechanisms—such as second-moment explosion in FedYogi and gradient starvation in small clients—advances theoretical understanding of limitations in federated adaptive optimization. Finally, the study confirms the superiority of federated learning over local models in data-siloed scenarios, with direct implications for privacy-preserving IDS design in edge-IoT environments.

Limitations of this work stem from the restricted client scale (n=3), which does not capture the dynamics of large-scale IoT networks with heterogeneous dropout and latency. Reliance on a single dataset (RT-IoT2022) may introduce bias toward specific attack patterns and limits generalization to zero-day or evolving threats. Additionally, the assumption of ideal synchronous communication overlooks real-world effects of stragglers, packet loss, and energy consumption on low-power IoT devices.

Future research should evaluate SCAFFOLD on 50–100 clients with simulated random dropout, network latency, and power profiling to assess system scalability under realistic conditions. Integrating personalization techniques such as FedPer or client clustering, alongside synthetic data augmentation via CTGAN or SMOTE-FL on minority clients, could further mitigate client drift. Developing hybrid FedAvg–SCAFFOLD strategies with dynamic switching based on drift estimation (e.g., via gradient cosine similarity) warrants exploration. Cross-dataset validation using CIC-IDS2018, UNSW-NB15, and Edge-IIoTset is necessary to test generalization across network topologies and attack variants. Finally, robustness testing against adversarial attacks (e.g., model poisoning, backdoors) and adaptation to concept drift in real-time data streams will strengthen the practical deployment of federated learning in dynamic IoT IDS environments.

5. Conclusion

This study presents a rigorous empirical benchmark of four advanced federated learning (FL) algorithms—FedAvg, SCAFFOLD, FedYogi, and AdaFedAdam—on the RT-IoT2022 dataset under extreme non-IID conditions (Dirichlet $\alpha = 0.01$, average JSD = 0.5677), simulating real-world IoT data silos with severe class imbalance and client heterogeneity. By integrating 10-fold cross-validation within the FL training loop and employing a multilayer neural network, the evaluation reveals SCAFFOLD as the most robust algorithm, achieving stable convergence from early rounds and superior final performance (Round 100: accuracy 0.7981, F1-score 0.7451, ROC-AUC 0.9396, log-loss 0.5882) through effective mitigation of client drift via control variables. In contrast, FedAvg demonstrates slow but reliable improvement (accuracy 0.6959), suitable for bandwidth-limited environments, while adaptive optimizers FedYogi and AdaFedAdam systematically fail due to second-moment instability and gradient starvation in sparse-data clients—exposing critical limitations in applying centralized adaptive methods to FL. Local models, confined to biased distributions, underscore the necessity of knowledge aggregation across clients. When compared to centralized IDS achieving near-perfect accuracy (up to 1.000), FL incurs a ~20% performance gap but guarantees explicit privacy by transmitting only model weights, aligning with GDPR and edge-IoT constraints. This work contributes the first validation of SCAFFOLD on an IoT IDS dataset under extreme heterogeneity, quantifies FedAvg convergence dynamics for practical deployment guidance, elucidates failure mechanisms of adaptive FL optimizers, and establishes a statistically robust, reproducible evaluation framework. These findings lay a foundation for privacy-preserving, scalable intrusion detection in resource-constrained IoT ecosystems, with future directions including large-scale client simulations, personalization techniques, and resilience against adversarial and evolving threats.

Acknowledgement

The authors thank the reviewers for their constructive comments that helped improve the quality of this paper. We also gratefully acknowledge the support from Telkom University (Surabaya Campus) for providing access to the laboratory facilities used to run and compile the federated learning experiments in this study.

References

[1] A. J. G. de Azambuja, T. Giese, K. Schützer, R. Anderl, B. Schleich, and V. R. Almeida, "Digital Twins in Industry 4.0 – Opportunities and challenges related to Cyber Security," *Procedia CIRP*, vol. 121, pp. 25–30, 2024, doi: 10.1016/j.procir.2023.09.225.

[2] M. Toussaint, S. Krima, and H. Panetto, "Industry 4.0 data security: A cybersecurity frameworks review," *J. Ind. Inf. Integr.*, vol. 39, p. 100604, 2024, doi: 10.1016/j.jii.2024.100604.

[3] S. Sai, M. Kanadia, and V. Chamola, "Empowering IoT with Generative AI: Applications, Case Studies, and Limitations," *IEEE Internet Things Mag.*, vol. 7, no. 3, pp. 38–43, 2024, doi: 10.1109/IOTM.001.2300246.

[4] C. Tagliaro, M. Komsic, A. Continella, K. Borgolte, and M. Lindorfer, "Large-Scale Security Analysis of Real-World Backend Deployments Speaking IoT-Focused Protocols," in *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, in RAID '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 561–578. doi: 10.1145/3678890.3678899.

[5] S. Sharma, V. Kumar, and K. Dutta, "Multi-objective optimization algorithms for intrusion detection in IoT networks: A systematic review," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 258–267, 2024, doi: 10.1016/j.iotcps.2024.01.003.

[6] B. Alotaibi, "A Survey on Industrial Internet of Things Security: Requirements, Attacks, AI-Based Solutions, and Edge Computing Opportunities," *Sensors*, vol. 23, no. 17, p. 7470, 2023, doi: 10.3390/s23177470.

[7] C. Guo, S. Wang, K. Yu, Y. Zhu, and X. Tao, "A Differential Game Method Against DDoS Attacks in IoT Botnets: Holistic and Dynamic Perspectives," *IEEE Internet Things J.*, vol. 12, no. 12, pp. 19414–19427, 2025, doi: 10.1109/JIOT.2025.3541852.

[8] S. Selvam and U. Maheswari Balasubramanian, "UASDAC: An Unsupervised Adaptive Scalable DDoS Attack Classification in Large-Scale IoT Network Under Concept Drift," *IEEE Access*, vol. 12, pp. 64701–64716, 2024, doi: 10.1109/ACCESS.2024.3397512.

[9] C. Xenofontos, I. Zografopoulos, C. Konstantinou, A. Jolfaei, M. K. Khan, and K.-K. R. Choo, "Consumer, Commercial, and Industrial IoT (In)Security: Attack Taxonomy and Case Studies," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 199–221, 2022, doi: 10.1109/JIOT.2021.3079916.

[10] A. Samuel and G. N. Edegbe, "A Systematic Review of Centralized and Decentralized Machine Learning Models: Security Concerns, Defenses and Future Directions," *NIPES - J. Sci. Technol. Res.*, vol. 6, no. 4, 2025, doi: 10.5281/zenodo.14681449.

[11] E. Jafarigol, T. B. Trafalis, T. Razzaghi, and M. Zamankhani, "Exploring Machine Learning Models for Federated Learning: A Review of Approaches, Performance, and Limitations," in *Dynamics of Disasters*, vol. 217, I. S. Kotsireas, A. Nagurney, P. M. Pardalos, S. W. Pickl, and C. Vogiatzis, Eds., in Springer Optimization and Its Applications, vol. 217. , Cham: Springer, 2024. doi: 10.1007/978-3-031-74006-0_4.

[12] J. Sophia, "Data Privacy Against Innovation or Against Discrimination?: The Case of the California Consumer Privacy Act (CCPA)," *Telemat. Inform.*, vol. 52, 2020, doi: 10.1016/j.tele.2020.101431.

[13] C. Prince, N. Omrani, and F. Schiavone, "Online privacy literacy and users' information privacy empowerment: the case of GDPR in Europe," *Inf. Technol. People*, vol. 37, no. 8, pp. 1–24, Jan. 2024, doi: 10.1108/ITP-05-2023-0467.

[14] M. Taufiq and A. S. Kenyo, "The Legal Protection of Personal Data in the Digital Era: A Comparative Study of Indonesian Law and the GDPR," *Int. J. Bus. Law Educ.*, vol. 6, no. 2, pp. 1260–1268, Aug. 2025, doi: 10.56442/ijble.v6i2.1178.

[15] B. S. Chaudhari, M. Zennaro, and S. Borkar, "LPWAN Technologies: Emerging Application Characteristics, Requirements, and Design Considerations," *Future Internet*, vol. 12, no. 3, 2020, doi: 10.3390/fi12030046.

[16] N. Ruminot, C. Estévez, V. D. P. Souto, R. D. Souza, and S. Montejo-Sánchez, "Improving the Reliability of Lightweight Blockchain LPWAN Transmission Schemes," *IEEE Sens. J.*, vol. 24, no. 17, pp. 28183–28195, 2024, doi: 10.1109/JSEN.2024.3428335.

[17] Z. B. Celik *et al.*, "Sensitive information tracking in commodity IoT," in *Proceedings of the 27th USENIX Conference on Security Symposium*, in SEC'18. USA: USENIX Association, 2018, pp. 1687–1704.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., in Proceedings of Machine Learning Research, vol. 54. PMLR, Apr. 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[19] M. R. A. Berkani *et al.*, "Advances in Federated Learning: Applications and Challenges in Smart Building Environments and Beyond," *Computers*, vol. 14, no. 4, p. 124, 2025, doi: 10.3390/computers14040124.

[20] A. Khraisat, A. Alazab, S. Singh, T. Jan, and A. Jr. Gomez, "Survey on Federated Learning for Intrusion Detection System: Concept, Architectures, Aggregation Strategies, Challenges, and Future Directions," *ACM Comput Surv*, vol. 57, no. 1, Oct. 2024, doi: 10.1145/3687124.

[21] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A Survey on Federated Learning for Resource-Constrained IoT Devices," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 1–24, 2022, doi: 10.1109/JIOT.2021.3095077.

[22] M. Adam and U. Baroudi, "Federated Learning for IoT: Applications, Trends, Taxonomy, Challenges, Current Solutions, and Future Directions," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 7842–7877, 2024, doi: 10.1109/OJCOMS.2024.3506214.

[23] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities," *IEEE Internet Things Mag.*, vol. 5, no. 1, pp. 24–29, 2022, doi: 10.1109/IOTM.004.2100182.

[24] H. Wang *et al.*, "Attack of the Tails: Yes, You Really Can Backdoor Federated Learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 16070–16084. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/b8ffa41d4e492f0fad2f13e29e1762eb-Paper.pdf

[25] L. Lyu *et al.*, "Privacy and Robustness in Federated Learning: Attacks and Defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8726–8746, 2024, doi: 10.1109/TNNLS.2022.3216981.

[26] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, Sep. 2021, doi: 10.1016/j.neucom.2021.07.098.

[27] J. Zhang *et al.*, "Adaptive Federated Learning on Non-IID Data With Resource Constraint," *IEEE Trans. Comput.*, vol. 71, no. 7, pp. 1655–1667, 2022, doi: 10.1109/TC.2021.3099723.

[28] L. Zhang, Y. Luo, Y. Bai, B. Du, and L.-Y. Duan, "Federated Learning for Non-IID Data via Unified Feature Learning and Optimization Objective

Alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 4420–4428.

[29] Z. Zhao *et al.*, "Federated Learning With Non-IID Data in Wireless Networks," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 3, pp. 1927–1942, 2022, doi: 10.1109/TWC.2021.3108197.

[30] A. Alamleh *et al.*, "Federated Learning for IoMT Applications: A Standardization and Benchmarking Framework of Intrusion Detection Systems," *IEEE J. Biomed. Health Inform.*, vol. 27, no. 2, pp. 878–887, 2023, doi: 10.1109/JBHI.2022.3167256.

[31] M. Almohaimeed and F. Albalwy, "Enhancing IoT Network Security Using Feature Selection for Intrusion Detection Systems," *Appl. Sci.*, vol. 14, no. 24, p. 11966, 2024, doi: 10.3390/app142411966.

[32] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," 2018, doi: 10.48550/ARXIV.1806.00582.

[33] G. Tuteja, A. R. Singh, G. R. Kumar, and M. H. Fallaah, "XGBoost-Based Anomaly Detection in IoT Networks Using the RT-IoT2022 Dataset," in *2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*, 2025, pp. 1–6. doi: 10.1109/OTCON65728.2025.11070793.

[34] G. Tuteja, A. R. Singh, and G. Ranjith Kumar, "Anomaly Detection in IoT Networks Using Random Forest and the RT-IoT2022 Dataset," in *2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*, 2025, pp. 1–6. doi: 10.1109/OTCON65728.2025.11071184.

[35] E. Çelik and M. K. Güllü, "Comparison of Federated Learning Strategies on ECG Classification," in *2023 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2023, pp. 1–4. doi: 10.1109/ASYU58738.2023.10296796.

[36] J. Liu, J. Huang, Y. Zhou, and others, "From distributed machine learning to federated learning: a survey," *Knowl. Inf. Syst.*, vol. 64, pp. 885–917, 2022, doi: 10.1007/s10115-022-01664-x.

[37] S. Liu, G. Yu, X. Chen, and M. Bennis, "Joint User Association and Resource Allocation for Wireless Hierarchical Federated Learning With IID and Non-IID Data," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 10, pp. 7852–7866, 2022, doi: 10.1109/TWC.2022.3162595.

[38] L. Ju, T. Zhang, S. Toor, and A. Hellander, "Accelerating Fair Federated Learning: Adaptive Federated Adam," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, pp. 1017–1032, 2024, doi: 10.1109/TMLCN.2024.3423648.

[39] J. Li, X. Tong, J. Liu, and L. Cheng, "An Efficient Federated Learning System for Network Intrusion Detection," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2455–2464, 2023, doi: 10.1109/JSYST.2023.3236995.

[40] P. Ruzafa-Alcázar *et al.*, "Intrusion Detection Based on Privacy-Preserving Federated Learning for the Industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1145–1154, 2023, doi: 10.1109/TII.2021.3126728.

[41] L. Lavaur, M.-O. Pahl, Y. Busnel, and F. Autrel, "The Evolution of Federated Learning-Based Intrusion Detection and Mitigation: A Survey," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 3, pp. 2309–2332, 2022, doi: 10.1109/TNSM.2022.3177512.

[42] B. Ghimire and D. B. Rawat, "Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8229–8249, 2022, doi: 10.1109/JIOT.2022.3150363.

[43] J. Cui, Y. Li, Q. Zhang, Z. He, and S. Zhao, "A Federated Learning Framework Using FedProx Algorithm for Privacy-Preserving Palmprint Recognition," in *Biometric Recognition*, S. Yu, W. Jia, X. Shu, X. Yuan, J. Gui, J. Tang, C. Shan, and Q. Liu, Eds., Singapore: Springer Nature Singapore, 2025, pp. 187–196. doi: https://doi.org/10.1007/978-981-96-1071-6_17.

[44] C. Mathew and P. Asha, "FedProx: FedSplit algorithm based federated learning for statistical and system heterogeneity in medical data communication," *J Internet Serv Inf Secur*, vol. 14, no. 3, pp. 353–370, 2024, doi: 10.58346/JISIS.2024.I3.021.

[45] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-Based Federated Learning for Securing Internet of Things: A Comprehensive Survey," *ACM Comput Surv*, vol. 55, no. 9, Jan. 2023, doi: 10.1145/3560816.

[46] S. Bates, T. Hastie, and R. Tibshirani, "Cross-Validation: What Does It Estimate and How Well Does It Do It?," *J. Am. Stat. Assoc.*, vol. 119, no. 546, pp. 1434–1445, 2024, doi: 10.1080/01621459.2023.2197686.

[47] L. A. Yates, Z. Aandahl, S. A. Richards, and B. W. Brook, "Cross validation for model selection: A review with examples from ecology," *Ecol. Monogr.*, vol. 93, no. 1, p. e1557, 2023, doi: https://doi.org/10.1002/ecm.1557.

[48] A. Sharma and H. Babbar, "Detecting Malicious Network Activities: Machine Learning-Based ARP Poisoning Detection on RT-IoT2022 Dataset," in *2024 Asian Conference on Intelligent Technologies (ACOIT)*, 2024, pp. 1–6. doi: 10.1109/ACOIT62457.2024.10939158.

[49] B. S and R. Nagapadma, "RT-IoT2022 Dataset." 2023. doi: 10.24432/C5P338.

[50] M. Chaudhary, L. Gaur, and A. Chakrabarti, "Detecting the Employee Satisfaction in Retail: A Latent Dirichlet Allocation and Machine Learning approach," in *2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, 2022, pp. 1–6. doi: 10.1109/ICCAKM54721.2022.9990186.

[51] H. Reguieg, M. E. Hanjri, M. E. Kamili, and A. Kobbane, "A Comparative Evaluation of FedAvg and Per-FedAvg Algorithms for Dirichlet Distributed Heterogeneous Data," in *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2023, pp. 1–6. doi: 10.1109/WINCOM59760.2023.10322899.

[52] A. T. Hutcheson and K. G. Brown, "Chi-Square," in *Statistics for Psychology Research*, Cham: Palgrave Macmillan, 2024. doi: 10.1007/978-3-031-60970-1_10.

[53] F. Nielsen, "Two Types of Geometric Jensen–Shannon Divergences," *Entropy*, vol. 27, no. 9, p. 947, 2025, doi: 10.3390/e27090947.

[54] L. Chen, W. Liu, Y. Chen, and W. Wang, "Communication-Efficient Design for Quantized Decentralized Federated Learning," *IEEE Trans. Signal Process.*, vol. 72, pp. 1175–1188, 2024, doi: 10.1109/TSP.2024.3363887.

[55] M. Ye, W. Shen, B. Du, E. Snezhko, V. Kovalev, and P. C. Yuen, "Vertical federated learning for effectiveness, security, applicability: A survey," *ACM Comput. Surv.*, vol. 57, no. 9, pp. 1–32, 2025, doi: 10.1145/3720539.

[56] G. Nilsson, "The Impact of Data Quality on Federated Versus Centralized Learning." , Department of Computer Science, 2024.

[57] D. Naik and N. Naik, "The Changing Landscape of Machine Learning: A Comparative Analysis of Centralized Machine Learning, Distributed Machine Learning and Federated Machine Learning," in *Advances in Computational Intelligence Systems. UKCI 2023*, vol. 1453, N. Naik, P. Jenkins, P. Grace, L. Yang, and S. Prajapat, Eds., in Advances in Intelligent Systems and Computing, vol. 1453. , Cham: Springer, 2024. doi:

10.1007/978-3-031-47508-5_2.

[58] S. Saha, A. Hota, A. K. Chattopadhyay, and others, "A Multifaceted Survey on Privacy Preservation of Federated Learning: Progress, Challenges, and Opportunities," *Artif. Intell. Rev.*, vol. 57, p. 184, 2024, doi: 10.1007/s10462-024-10766-7.

[59] A. Yazdinejad, A. Dehghantanha, G. Srivastava, H. Karimipour, and R. Parizi, "Hybrid Privacy Preserving Federated Learning Against Irregular Users in Next-Generation Internet of Things," *J. Syst. Archit.*, vol. 148, p. 103088, 2024, doi: 10.1016/j.sysarc.2024.103088.

[60] E. Price, "Federated Learning for Privacy-Preserving Edge Intelligence: A Scalable Systems Perspective," *J. Comput. Sci. Softw. Appl.*, vol. 5, no. 5, 2025, doi: 10.5281/zenodo.15381925.

[61] K. Hu, S. Gong, Q. Zhang, and others, "An overview of implementing security and privacy in federated learning," *Artif. Intell. Rev.*, vol. 57, p. 204, 2024, doi: 10.1007/s10462-024-10846-8.

[62] S. Sebbio, G. Morabito, A. Catalfamo, L. Carnevale, and M. Fazio, "Federated Learning on Raspberry Pi 4: A Comprehensive Power Consumption Analysis," in *Proceedings of the 16th IEEE/ACM International Conference on Utility and Cloud Computing (UCC '23)*, New York, NY, USA: Association for Computing Machinery, 2024, pp. 1–6. doi: 10.1145/3603166.3632545.

[63] L. Ridolfi, D. Naseh, S. S. Shinde, and D. Tarchi, "Implementation and Evaluation of a Federated Learning Framework on Raspberry PI Platforms for IoT 6G Applications," *Future Internet*, vol. 15, no. 11, p. 358, 2023, doi: 10.3390/fi15110358.

[64] M. Ramadan, M. Ali, S. Y. Khoo, and M. Alkhedher, "Federated Learning and TinyML on IoT Edge Devices: Challenges, Advances, and Future Directions," *ICT Express*, vol. 11, 2025, doi: 10.1016/j.icte.2025.06.008.

[65] M. H. Bhavsar, Y. B. Bekele, K. Roy, J. C. Kelly, and D. Limbrick, "FL-IDS: Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT," *IEEE Access*, vol. 12, pp. 52215–52226, 2024, doi: 10.1109/ACCESS.2024.3386631.

[66] B. Casella and S. Fonio, "Architecture-Based FedAvg for Vertical Federated Learning," in *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing*, in UCC '23. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3603166.3632559.

[67] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *Proceedings of the 37th International Conference on Machine Learning*, H. D. III and A. Singh, Eds., in Proceedings of Machine Learning Research, vol. 119. PMLR, Jul. 2020, pp. 5132–5143. [Online]. Available: https://proceedings.mlr.press/v119/karimireddy20a.html

[68] S. Reddi *et al.*, "Adaptive Federated Optimization." 2021. [Online]. Available: https://arxiv.org/abs/2003.00295

[69] J. Kumarappan, E. Rajasekar, S. Vairavasundaram, and others, "Federated Learning Enhanced MLP–LSTM Modeling in an Integrated Deep Learning Pipeline for Stock Market Prediction," *Int. J. Comput. Intell. Syst.*, vol. 17, p. 267, 2024, doi: 10.1007/s44196-024-00680-9.

[70] H. Li, W. Ouyang, and X. Wang, "Multi-Bias Non-linear Activation in Deep Neural Networks," in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., in Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, Jun. 2016, pp. 221–229. [Online]. Available: https://proceedings.mlr.press/v48/lia16.html

[71] S. Sharma, S. Sharma, and A. Athaiya, "ACTIVATION FUNCTIONS IN NEURAL NETWORKS," *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, pp. 310–316, May 2020, doi: 10.33564/IJEAST.2020.v04i12.054.

[72] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, "Federated Learning With Spiking Neural Networks," *IEEE Trans. Signal Process.*, vol. 69, pp. 6183–6194, 2021, doi: 10.1109/TSP.2021.3121632.

[73] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting Weighted Aggregation in Federated Learning with Neural Networks," in *Proceedings of the 40th International Conference on Machine Learning*, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., in Proceedings of Machine Learning Research, vol. 202. PMLR, Jul. 2023, pp. 19767–19788. [Online]. Available: https://proceedings.mlr.press/v202/li23s.html

[74] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in *Artificial Intelligence Application in Networks and Systems*, R. Silhavy and P. Silhavy, Eds., Cham: Springer International Publishing, 2023, pp. 15–25. doi: https://doi.org/10.1007/978-3-031-35314-7_2.

[75] J. Terven, D. M. Cordova-Esparza, and J. A. Romero-González, "A Comprehensive Survey of Loss Functions and Metrics in Deep Learning," *Artif. Intell. Rev.*, vol. 58, p. 195, 2025, doi: 10.1007/s10462-025-11198-7.

[76] A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," *Expert Syst. Appl.*, vol. 244, p. 122778, Jun. 2024, doi: 10.1016/j.eswa.2023.122778.

[77] B. S. Sharmila and R. Nagapadma, "Quantized Autoencoder (QAE) Intrusion Detection System for Anomaly Detection in Resource-Constrained IoT Devices Using RT-IoT2022 Dataset," *Cybersecurity*, vol. 6, p. 41, 2023, doi: 10.1186/s42400-023-00178-5.

[78] B. S. Sharmila, B. M. Nandini, S. S. Kavitha, and A. Srivatsa, "Performance Evaluation of Parametric and Non-Parametric Machine Learning Models Using Statistical Analysis for RT-IoT2022 Dataset," *J. Sci. Ind. Res. Comput. Sci. Commun. Inf. Technol.*, vol. 83, no. 8, 2024.

[79] A. Hamdan, M. Tahboush, M. Adawy, T. Alwada'n, and S. Ghwanmeh, "Feature Reduction and Anomaly Detection in IoT Using Machine Learning Algorithms," *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 16, no. 1, 2025, doi: 10.14569/IJACSA.2025.0160146.

[80] A. Prasad, W. M. Alenazy, N. Ahmad, and others, "Optimizing IoT Intrusion Detection with Cosine Similarity Based Dataset Balancing and Hybrid Deep Learning," *Sci. Rep.*, vol. 15, p. 30939, 2025, doi: 10.1038/s41598-025-15631-3.