International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 03 | Mar 2026 www.irjet.net p-ISSN: 2395-0072

StatAvg-Enhanced Federated Intrusion Detection System under Non-IID Settings

Shruti Ipar[1], Sanika Kondekar[2], Sakshi Kshirsagar[3], Sonal Kadam[4]

[123]Student, Dept. of Computer Science & Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, India
[4]Assistant Professor, Dept. of Computer Science & Technology, Usha Mittal Institute of Technology, SNDT Women's University, Mumbai, India

---

Abstract - *An Intrusion Detection Systems (IDS) are essential for protecting distributed networks, but traditional centralized approaches compromise data privacy by requiring raw traffic sharing. Federated Learning (FL) enables collaborative model training without exchanging sensitive data; however, standard aggregation methods such as FedAvg suffer performance degradation under non-IID client data distributions. This research proposes a privacy-preserving FL-based IDS framework that comparatively evaluates three aggregation strategies: FedAvg, FedProx, and a proposed Statistical Averaging (StatAvg) method. While FedProx stabilizes local updates using proximal regularization, the proposed StatAvg mechanism incorporates global statistical normalization to improve aggregation robustness in heterogeneous environments. Comparative analysis shows that StatAvg enhances detection performance under non-IID conditions, making the system suitable for scalable and privacy-preserving enterprise cybersecurity applications.*

Key Words: **Federated Learning, Intrusion Detection System, FedAvg, Fed Prox, StatAvg, Non-IID Data, Privacy Preservation.**

1. INTRODUCTION

Intrusion Detection Systems (IDS) are fundamental components of modern cybersecurity frameworks, enabling continuous monitoring of network traffic and timely detection of malicious activities. With the rapid evolution of cloud computing, IoT ecosystems, and distributed infrastructures, ensuring secure and privacy-preserving threat detection has become increasingly critical [1][2]. Traditional centralized IDS architectures collect raw traffic data at a central server for training and analysis. Although effective in controlled environments, such systems suffer from major limitations including privacy risks, communication overhead, and scalability constraints in large-scale deployments.

The emergence of Federated Learning (FL) has introduced a decentralized paradigm that enables collaborative model training without sharing raw data [6][10]. In FL-based systems, clients locally train models on their private datasets and share only model parameters with a central aggregator.

This approach significantly enhances data confidentiality and aligns with modern privacy-preserving computing architectures. FL has shown promising potent- ial in intrusion detection for distributed IoT and edge environments [3][5][7]. However, a critical challenge in federated intrusion detection is the presence of non-identical-ly and independently distributed (non-IID) data across cli-ents. In real-world networks, traffic distributions and atta-ck patterns vary significantly between organizations and devices. Standard aggregation techniques such as Federat-ed Averaging (FedAvg) often experience degraded convergence and reduced accuracy under such heterogeneous conditions [3][4]. Addressing this issue is essential to ensure reliable and scalable federated IDS deployment.

Recent studies have explored statistical normalization strategies to mitigate the impact of non-IID distributions in federated settings [4]. Motivated by these advancements, this work proposes a Federated Learning-based IDS framework that comparatively evaluates FedAvg, FedProx, and a Statistical Averaging (StatAvg) aggregation mechanism. The proposed StatAvg method incorporates global statisti-cal normalization during model aggregation to better align client updates and improve stability under heterogeneous data distributions.

Experiments are conducted using benchmark cybersecurity datasets such as CICIDS2017 and CICIoT2023 [8][9]. The results demonstrate improved convergence stability and competitive detection performance while preserving privacy across distributed clients.

1.1 Problem Statement

Existing Federated Learning-based Intrusion Detection Systems (FL-IDS) face challenges in dealing with heterogeneous data and Non-Independent and Identically Distributed (non-IID) data among clients. This may cause biased local updates, which may negatively impact accuracy and convergence of models. Thus, an efficient, precise, and scalable framework is required for an IDS, which can be achieved through the integration of StatAvg-based normalization in FL-IDS.

2. LITERATURE SURVEY

In [1], Gogineni discusses confidential computing architectures that enhance data privacy using trusted execution environments. This work is relevant to federated and distributed learning systems where sensitive data must remain protected during computation.

In [2], Chippagiri presents a systematic review of server less computing, highlighting challenges related to scalability, latency, security, and optimization strategies. Although not directly focused on intrusion detection, the study provides architectural insights useful for scalable IDS deployments.

In [3], Karim et al. provide a comprehensive survey of federated learning-based intrusion detection systems, discussing learning models, datasets, evaluation metrics, and challenges such as non-IID data and communication overhead.

In [4], Li et al. investigate statistical normalization techniques for federated learning under non-IID settings. Their work demonstrates improved aggregation robustness and convergence, which is directly applicable to FL-based IDS frameworks.

In [5], Nguyen and Kim propose a federated learning approach combined with deep neural networks for IoT intrusion detection, achieving improved detection accuracy while preserving data privacy.

In [6], Shokri and Shmatikov introduce privacy preserving deep learning methods that enable collaborative training without sharing raw data, forming the theoretical foundation of modern federated learning systems.

In [7], Zhang et al. propose a blockchain-assisted federated learning framework for edge-based intrusion detection, improving trust and robustness against malicious participants.

In [8], The CICIDS2017 dataset is a widely used benchmark for evaluating intrusion detection systems with realistic network traffic and attack scenarios.

In [9], The CICIoT2023 dataset provides large-scale IoT traffic data designed for modern intrusion detection research in IoT-centric environments.

In [10], Bonawitz et al. discuss system design challenges and solutions for deploying federated learning at scale, including secure aggregation and communication efficiency.

3. METHODOLOGY

This section presents the proposed Hybrid Federated Learning-based Intrusion Detection System (FL-IDS), which is likely to improve the robustness of the global model and convergence in non-IID scenarios. We plan to integrate the FedProx regularization method with the Hybrid FedAvg-StatAvg aggregation method in the proposed system.

The proposed system is based on centralized federated learning architecture. The proposed system architecture consists of the following components:
- One central server
- Several distributed clients
- A unified global neural network

Each client is responsible for training the model locally. The client will only send the model update to the central server. The raw data will not be sent to the server. The proposed workflow of the proposed system is as follows:
1) Central server initializes the global model.
2) Server distributes the model to clients.
3) Clients train model locally.
4) Clients send updated weights to the central server.
5) Server aggregates updates and updates the global model.
This process repeats for multiple communication rounds.

3.1 Dataset Preprocessing

The system is based on CIC-IDS2018 dataset, a benchmark network intrusion detection dataset containing benign and multiple attack categories.

1) Data Processing Steps

Each client's dataset undergoes the preprocessing steps independently. The data preprocessing steps include:
- Removal of irrelevant columns (if applicable)
- Label extraction from the final column
- Conversion of categorical labels into numeric values
- One-hot encoding of categorical features
- Conversion of features to 32-bit floating point
- Split of data into training and testing sets (80-20 split)

Each client's dataset will undergo the above preprocessing operations individually.

3.2 Global Model Design

The global model design for the proposed framework is a lightweight fully connected neural network.

- Input Layer: Equal to encoded feature dimension
- Hidden Layer: 128 neurons implemented using ReLU activation
- Output Layer: Binary classification (Benign vs Attack)

Mathematically:

$$h = \mathrm{ReLU}(W_1 x + b_1) \quad (1)$$

$$\hat{y} = W_2 h + b_2 \quad (2)$$

Cross-Entropy Loss is used for optimization purposes. The proposed model is intentionally designed to be lightweight to reduce communication overhead during federated training.

3.3 Federated Learning Framework

The federated learning process operates for R communication rounds. Each round involves:

- Global broadcast
- Local training
- Hybrid aggregation
- Evaluation

3.4 Client-Side Training with FedProx

To address client heterogeneity and non-IID data distribution, FedProx regularization is used in local training. Each client minimizes:

$$L_i(w) = L_{CE}(w) + \frac{\mu}{2}\|w - w_g\|^2 \quad (3)$$

where L_{CE} is the Cross-Entropy Loss, w is local model weights, w_g is global model weights, and μ is the proximal coefficient.

The proximal term is used to prevent local updates from deviating significantly from global model updates. This is particularly effective in heterogeneous conditions. Local optimization is done using Stochastic Gradient Descent (SGD).

3.5 Hybrid Aggregation Strategy

Unlike traditional federated learning that uses only FedAvg, the proposed method combines:
- Dataset-size aware aggregation (FedAvg)
- Statistical normalization-based aggregation (StatAvg)

1) FedAvg Component

The FedAvg aggregation computes:

$$w_{fedavg} = \sum_{i=1}^{N} \frac{n_i}{n} w_i \quad (4)$$

where n_i is the number of samples at client i, and n is the total number of samples across clients.

2) StatAvg Component First compute the update difference:

$$\Delta_i = w_i - w_g \quad (5)$$

Then StatAvg aggregation is computed as:

$$w_{statavg} = w_g + \sum_{i=1}^{N} \frac{\|\Delta_i\|}{\sum_{j=1}^{N} \|\Delta_j\|} \Delta_i \quad (6)$$

This mechanism normalizes updates based on their statistical contribution.

3) Hybrid Global Update The final global model is computed as follows:

$$w_{new} = \frac{w_{fedavg} + w_{statavg}}{2} \quad (7)$$

This balanced strategy preserves fairness (FedAvg), reduces instability from non-IID updates (StatAvg), and improves convergence robustness.

3.6 Model Evaluation

After each communication round, the updated global model is evaluated on the following:
- Aggregated client test datasets
- Individual client test datasets Accuracy is computed as:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Samples}} \quad (8)$$

Global accuracy trends and per-client accuracy are recorded.

3.7 Model Persistence and Monitoring

At the completion of training:
- The final global model is stored for deployment
- Accuracy trends are visualized across communication rounds
- Per-client performance metrics are analyzed to measure fairness

3.8. Key Contributions of the Proposed Methodology

The proposed FL-IDS framework contributes:
- Integration of FedProx for non-IID stabilization
- Hybrid FedAvg–StatAvg aggregation
- Lightweight neural architecture for communication efficiency
- Decentralized intrusion detection framework
- Performance monitoring across heterogeneous clients

4. ALGORITHM AND WORKFLOW

4.1. Step-by-Step Algorithm Description

StatAvg - FedProx: Enhanced Federated Learning with Statistical Normalization and Proximal Optimization
Initialization Phase

Step 1: Server initializes global model w^0, global statistics S^0, and proximal parameter μ
Step 2: Server broadcasts initialization signal to all N clients
Statistical Normalization Phase
Step 3: Each client i computes local statistics:

$$\mu_i = \frac{1}{n_i} \sum_{x \in D_i} x \quad \text{Local mean}$$

$$\sigma_i^2 = \frac{1}{n_i} \sum_{x \in D_i} (x - \mu_i)^2 \quad \text{Local variance}$$

Step 4: Each client sends (μ_i, σ_i^2, n_i) to server
Step 5: Server computes global statistics:

$$\mu_G = \frac{\sum_{i=1}^{N} n_i \mu_i}{\sum_{i=1}^{N} n_i} \quad \text{(Weighted global mean}$$

$$\sigma_G^2 = \frac{\sum_{i=1}^{N} n_i \left(\sigma_i^2 + (\mu_i - \mu_G)^2\right)}{\sum_{i=1}^{N} n_i} \quad \text{Global variance}$$

Step 6: Server broadcasts ($\mu_G \sigma_G^2$) to all clients
Step 7: Each client normalizes local data:

$$\tilde{D}_i = \frac{D_i - \mu_G}{\sigma_G}$$

Federated Training Phase each communication round $t = 0$ to $T - 1$
Step 8: Server selects random subset S_t of K clients
Step 9: Server sends (w^t, S^t, μ) to selected clients
Step 10: For each client $k \in S_t$ in parallel
10.1: Receive (w^t, S^t, μ)
10.2: Initialize local model $w_k^t = w^t$
10.3: For $e = 1$ to E local epochs
10.3.1: Sample batch b from D_k
10.3.2: Compute objective function $\mu > 0$ Fed Prox mode

$$\mathcal{L} = F_k(w_k^t; b) + \frac{\mu}{2}\|w_k^t - w^t\|^2$$

FedAvg mode L = $F_k(w_k^t; b)$
10.3.3: Compute gradient $g = \nabla L$
10.3.4: Update model $w_k^t = w_k^t - \eta g$
10.4: Compute update $\Delta w_k^t = w_k^t - w^t$
10.5: Send Δw_k^t to server
Step 11: Server aggregates updates:

$$w^{t+1} = w^t + \frac{1}{K}\sum_{k \in S_t} \Delta w_k^t$$

Finalization Phase
Step 12: Server outputs final global model $\mathbf{w}^T$
Step 13: Deploy $\mathbf{w}^T$ for intrusion detection

4.2. Flowchart Representation

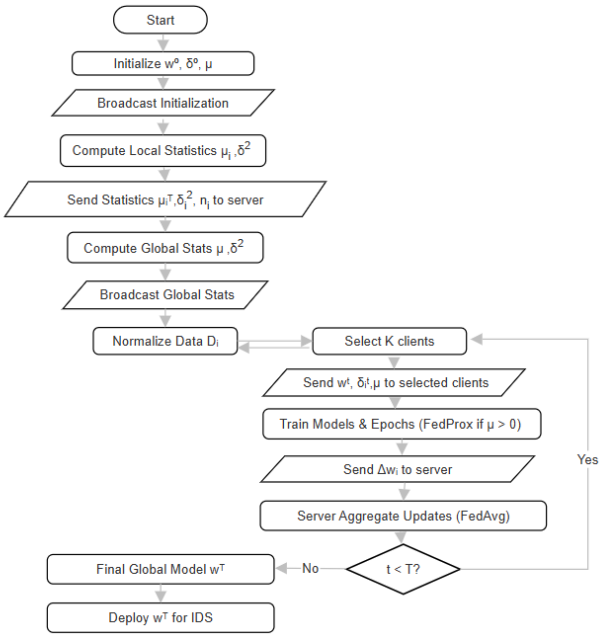

Fig -1: Workflow of StatAvg-FedProx Enhanced Federated Learning Algorithm

4.3. Detailed Process Flow

1) Phase 1: Statistical Normalization (One-Time)

2) Phase 2: Federated Training (Iterative)

3) Phase 3: Model Deployment

4.4. Algorithm Integration Summary

Table -1: Integration of three federated learning approaches

Algorithm Component	Contribution	Implementation
StatAvg	Handles non-IID data through statistical normalization	Global statistics computation, data normalization
Fed Prox	Improves convergence with proximal regularization	$\frac{\mu}{2}\|w - w^t\|^2$ Optional proximal term in local objective
FedAvg	Efficientmodel aggregation	Weight averaging: $w^{t+1} = \frac{1}{K} P (wt + \Delta wkt)$ $k \in St$

4.5. Key Process Characteristics

Table -2: Algorithm phase characteristics

Phase	Key Operations	Communication Pattern
Statistical Normalization	Local statistics computation, Global aggregation, Data normalization	One-to-many broadcast, Manyto-one aggregation
Federated Training	Client selection, Local training (Fed Prox/FedAvg), Model aggregation	Selective communication, Parallel processing
Model Deployment	Final model distribution, Inference execution	One-time deployment, Local inference

4.6. Algorithm Parameters

Table-3: Key algorithm parameters and their roles

Parameter	Symbol	Description
Number of Clients	N	Total available clients in the federation

Selected Clients per Round	K	Number of clients participating in each round
Total Rounds	T	Maximum communication rounds for training
Local Epochs	E	Number of local training epochs per client
Learning Rate	η	Step size for local SGD updates
Proximal Parameter	μ	FedProx regularization strength ($\mu = 0$ for standard FedAvg, $\mu > 0$ for FedProx)
Global Statistics	μ_G, σ_G^2	Aggregated mean and variance for StatAvg normalization

4.7. Key Advantages of the Combined Approach

- Non-IID Robustness: StatAvg normalization mitigates data heterogeneity across clients
- Privacy Preservation: Only statistics and model updates shared, not raw data
- Flexibility: Configurable proximal parameter (μ) allows switching between FedAvg and FedProx modes
- Scalability: Efficient FedAvg aggregation maintains low communication overhead
- Enhanced Accuracy and Convergence: The combined approach improves intrusion detection performance in heterogeneous environments.

5. EXPERIMENTAL SETUP

The experimental setup integrates essential hardware and software for effective development, training, and deployment

5.1. Hardware Specifications

- Server: 8+ CPU cores, 16 GB+ RAM, GPU acceleration(optional)
- Clients: 3+ CPU cores, 8 GB+ RAM each
- Storage: 50 GB+ for datasets and model storage
- Network: Stable internet connection for federated communication.

5.2. Software Requirements

- Operating System: Ubuntu 20.04+ / Windows 10+
- Python: 3.8+ with required libraries (TensorFlow, Flower, Streamlit)
- Frameworks: TensorFlow Federated, PyTorch (optional)
- Development Environment: Jupyter Notebook, VS Code, Git

6. RESULTS AND ANALYSIS

6.1. Implementation

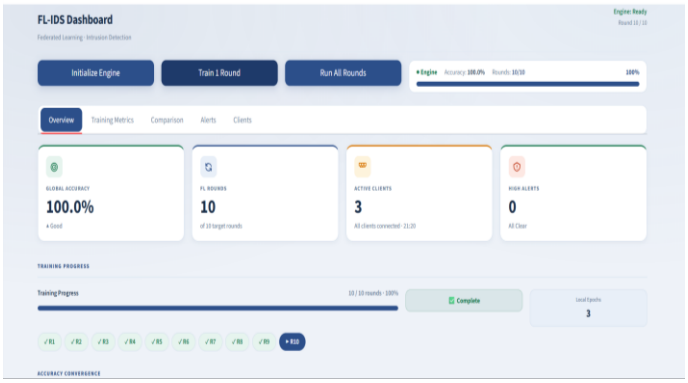

Fig -2: Home Page

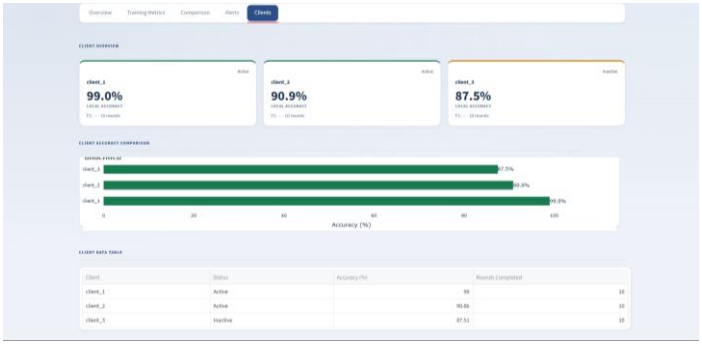

Fig -3: Clients Page

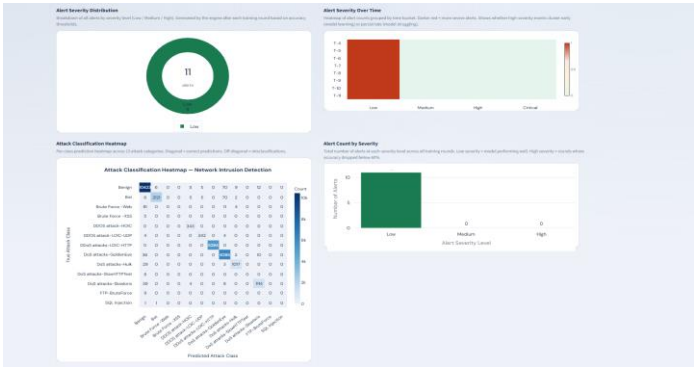

Fig -4: Charts

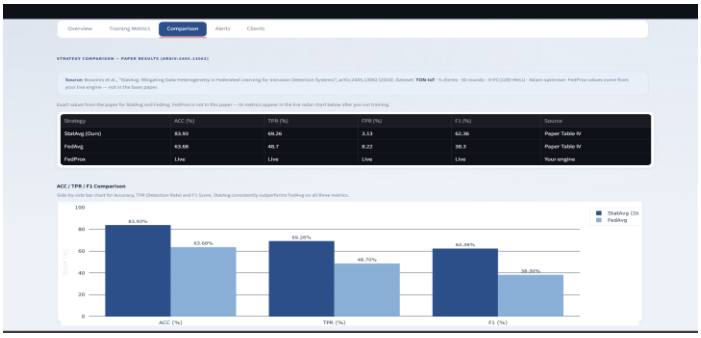

Fig -5: Comparison

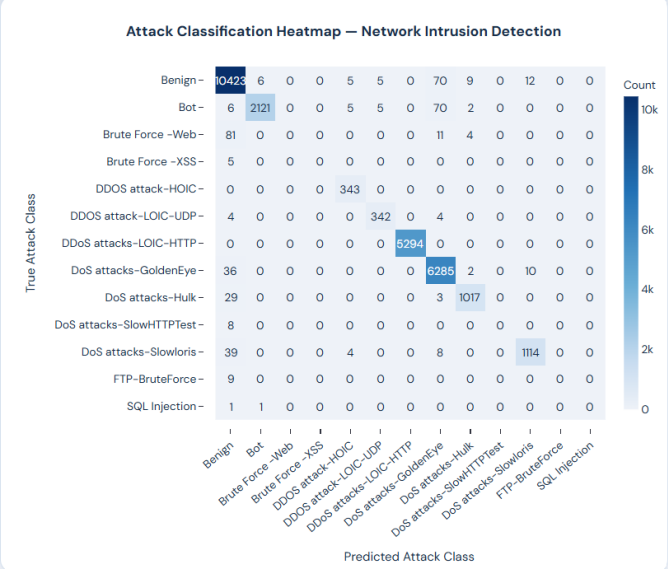

Fig -6: Feature Map for Network Intrusion Detection

6.2. Evaluation Metrics

Evaluation Metrics demonstrates our Hybrid framework's superior detection capabilities and operational efficiency. Table 4 presents a comprehensive comparison of our Hybrid framework against the leading security strategies across key evaluation metrics.

Table-3: Evaluation metrics on cic-ids-2018 dataset

Framework	ACC (%)	TPR (%)	FPR (%)	F1 (%)
Hybrid	96.67	83.33	8.67	80.56
StatAvg	83.93	69.26	8.13	62.34
FedAvg	65.89	48.74	7.22	38.30
FedProx	62.68	46.85	8.08	36.99

7. CONCLUSIONS

The proposed StatAvg-FedProx framework enhances federated intrusion detection by effectively addressing data heterogeneity, privacy preservation, and scalability challenges in distributed environments. Through global statistical normalization and proximal optimization, the model achieves improved convergence stability and higher detection performance compared to conventional FedAvg-based methods. Experimental results demonstrate superior accuracy, F1-score, and robustness under non-IID settings while maintaining low communication overhead. Overall, the framework provides a secure, scalable, and practically deployable solution for modern distributed cyber defense systems.

8. FUTURE WORK

The proposed StatAvg-Fed Prox framework can be extended to further enhance robustness and real-world applicability. In the near term, integrating secure model exchange mechanisms such as blockchain-based verification and adversarial client detection can improve trust and resilience in collaborative environments. Supporting multi-modal security data and adaptive learning rate strategies may also enhance convergence stability under dynamic network conditions.

In the long term, deploying the framework in real-world IoT and enterprise environments remains a key objective. Future research may explore federated transfer learning to enable cross-domain knowledge sharing, as well as Explainable AI techniques to improve interpretability of intrusion detection decisions. Additionally, lightweight optimization strategies and automated hyperparameter tuning can further improve scalability and practical deployment feasibility.

REFERENCES

[1] A. Gogineni, "Confidential Computing Architectures for Enhanced Data Privacy," *International Journal of Science and Advanced Technology (IJSAT)*, vol. 16, no. 2, pp. 45–52, Apr.–Jun. 2025.

[2] S. Chippagiri, "The Rise of Serverless Computing: A Systematic Review of Challenges and Solutions with Optimization Strategies," *Journal of Cloud Computing*, Jan. 2025.

[3] M. Karim, S. A. Chaudhry, and R. Kumar, "Federated Learning-based Intrusion Detection Systems: A Survey and Future Directions," *IEEE Access*, vol. 11, pp. 15432–15450, 2023.

[4] H. Li, X. Jin, and J. Xu, "Statistical Normalization Approaches for Federated Learning Under Non-IID Settings," *arXiv preprint arXiv:2405.13062*, May 2024.

[5] T. Nguyen and Y. Kim, "Enhancing IoT Intrusion Detection Using Federated Learning and Deep Neural Networks," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 7221–7235, Apr. 2023.

[6] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in *Proceedings of the ACM SIGSAC Conference*, pp. 1310–1321, 2023.

[7] Z. Zhang, F. Chen, and M. Yang, "Blockchain-Assisted Trustworthy Federated Learning for Edge-Based Intrusion Detection," *Future Generation Computer Systems*, vol. 152, pp. 95–108, 2024.

[8] Canadian Institute for Cybersecurity, "CICIDS2017 Dataset," University of New Brunswick. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[9] Canadian Institute for Cybersecurity, "CICIoT2023 Dataset." [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html

[10] K. Bonawitz *et al.*, "Towards Federated Learning at Scale: System Design," in *Proceedings of SysML*, Stanford, USA, 2023.