

From Models to Network Topologies: A Topology Inference Attack in Decentralized Federated Learning

Chao Feng^{a,*}, Yuanzhe Gao^a, Alberto Huertas Celdrán^{a,b}, Gérôme Bovet^c and Burkhard Stiller^a

^aCommunication Systems Group, Department of Informatics, University of Zurich, CH-8050 Zürich, Switzerland

^bDepartment of Information and Communications Engineering, University of Murcia, 30100–Murcia, Spain

^cCyber-Defence Campus, armasuisse Science & Technology, CH-3602 Thun, Switzerland

Abstract. Federated Learning (FL) is widely recognized as a privacy-preserving Machine Learning paradigm due to its model-sharing mechanism that avoids direct data exchange. Nevertheless, model training leaves exploitable traces that can be used to infer sensitive information. In Decentralized FL (DFL), the topology, defining how participants are connected, plays a crucial role in shaping the model’s privacy, robustness, and convergence. However, the topology introduces an unexplored vulnerability: attackers can exploit it to infer participant relationships and launch targeted attacks. This work uncovers the hidden risks of DFL topologies by proposing a novel Topology Inference Attack that infers the topology solely from model behavior. A taxonomy of topology inference attacks is introduced, categorizing them by the attacker’s capabilities and knowledge. Practical attack strategies are designed for various scenarios, and experiments are conducted to identify key factors influencing attack success. The results demonstrate that analyzing only the model of each node can accurately infer the DFL topology, highlighting a critical privacy risk in DFL systems. These findings offer insights for improving privacy preservation in DFL environments.

1 Introduction

Federated Learning (FL) has emerged as a novel framework for enabling privacy-preserving Machine Learning (ML), facilitating collaborative model training among distributed clients without the necessity of sharing raw data [22]. Conventional FL systems are predicated on a centralized architecture, referred to as Centralized FL (CFL), wherein a central entity is responsible for collecting, aggregating, and redistributing models to the clients. Nevertheless, this centralized architecture presents several challenges, such as processing bottlenecks and the single points of failure risk [2]. In response to these limitations, Decentralized FL (DFL) has been introduced, wherein model training and aggregation occur locally, and communication between nodes relies on a peer-to-peer (P2P) network. This approach eliminates dependency on a central server, mitigating risks associated with single points of failure [1].

Although FL, both centralized and decentralized, protect raw data privacy through its unique model-sharing mechanism, the training process inevitably leaves traces in the models. These traces can be exploited by malicious actors, potentially leading to inference attacks, including membership inference attacks, property inference attacks, and attribute inference attacks [18]. Previous studies have illustrated

that such attacks pose a considerable risk to privacy within FL systems by elucidating sensitive information based on the behavior of the models [5]. While much of the existing work focuses on CFL, a notable research gap exists in exploring information leakage in DFL.

Overlay network topology defines how participants are interconnected in an FL system. In CFL, the client-server architecture enforces a fixed, star-shaped topology. However, DFL leverages the P2P network, enabling flexible node connections to form diverse topologies. Existing studies demonstrate that topology significantly influences DFL models’ robustness and privacy-preserving capabilities [7]. From a security perspective, exposure of the DFL network topology significantly facilitates malicious attacks. For example, in poisoning attack scenarios, topology awareness allows adversaries to pinpoint critical nodes, such as the central node in star topologies, thereby enabling them to amplify the poisoning intensity and enhance its propagation effectiveness [7]. Similarly, for network-level attacks, knowledge of key nodes within the DFL topology can lead to more targeted and effective Distributed Denial-of-Service (DDoS) attacks. Thus, topology information must be regarded as a critical asset, both in terms of optimizing DFL training performance and ensuring system security. However, theoretical analyses on how network topology affects model convergence remain sparse. Moreover, there is limited research investigating potential information leakage risks for this sensitive overlay topology data and strategies to safeguard this critical asset.

Therefore, this paper improves this research gap by proposing a novel overlay topology inference attack targeting DFL systems. The proposed attack leverages the behavioral traces generated by models to uncover sensitive overlay topologies. The main contributions of this work are as follows: (i) the formulation of a novel taxonomy of topology inference attacks, classifying these attacks based on the attacker’s capabilities and knowledge; (ii) the development of practical strategies tailored to various attack types, accompanied by a quantitative analysis of critical factors that influence the efficiency of these attacks¹; (iii) rigorous experimental evaluations conducted across a range of datasets and real-world network topology configurations to empirically validate the proposed attack strategies; and (iv) the provision of insights aimed at informing the design of effective defensive mechanisms to safeguard sensitive information within DFL systems.

* Corresponding Author. Email: cfeng@ifi.uzh.ch.

¹ Code available at: <https://github.com/luke-feng/TIA>

2 Background and Related Work

This section overviews inference attacks, encompassing their classifications and potential attack surfaces. Since there is a lack of research directly related to topology inference attacks on DFL, this paper examines topology inference studies conducted in other domains, including communication systems and social networks.

2.1 Inference Attacks

Inference attacks exploit ML models to extract sensitive information without direct access to the underlying data [25]. By analyzing model behaviors or outputs, adversaries can infer properties of the data, model parameters, or system structure. As FL grows in adoption for its privacy-preserving capabilities, it also faces increased vulnerability to such attacks [32]. Inference attacks vary in objectives and strategies, with common types including:

- **Membership Inference Attack:** Determines whether a specific data sample was part of the training dataset, often using shadow models or prediction confidence scores to differentiate between training and unseen data [28]. This is critical in sensitive domains like healthcare or finance.
- **Model Inversion Attack:** Reconstructs input data or infers its properties by iteratively optimizing inputs to match observed model outputs, posing risks in applications involving sensitive information like facial recognition or medical data [9].
- **Property Inference Attack:** Infers characteristics of the training data, such as demographic distributions, by analyzing model updates or outputs. In FL, this can involve monitoring node-specific behaviors to deduce shared properties of local datasets [10].
- **Attribute Inference Attack:** Targets specific attributes of training data, often by injecting malicious samples to influence the model’s learning process and make it vulnerable to data leakage [12].

These attacks not only compromise users’ sensitive information and undermine data security but also erode trust in FL’s privacy preserving mechanisms. Users may hesitate to participate in the training process, undermining the overall effectiveness of the FL system. However, existing research primarily focuses on CFL, with limited exploration of privacy leakage in DFL. Moreover, these studies have not adequately addressed the critical role of DFL’s overlay topology.

2.2 Topology Inference

Topology inference is explored in various research domains, such as identifying source–destination paths in communication systems and uncovering interconnections among nodes in social networks.

2.2.1 Communication Systems

In communication systems, most protocols are designed to avoid ring formation, resulting in tree-like topologies. Many topology inference studies commonly assume a single path between nodes [3]. A widely used approach involves leveraging ICMP-based protocols, such as the traceroute, to map network topologies [19]. Tomography-based metrics, like packet loss rates, further aid in deducing a system’s topology: by observing loss rates from a source node to a target node, one can estimate the likelihood of a direct connection and thereby reconstruct the network [4]. Additionally, by measuring the timestamps of sent and received packets, it is possible to determine the number of forwarding hops, offering deeper insight into the overall network configuration [17].

2.2.2 Social Networks

Social networks generally refer to the social structure formed by individuals in society connected by a specific relationship. In social networks, user interactions (e.g., likes, retweets) serve as signals for topology modeling and inference by capturing how information diffuses across the network [11]. However, this behavioral data is often incomplete, especially in large-scale settings, necessitating methods like noisy sparse subspace clustering to infer network structures from partial observations [30]. Meanwhile, social networks are dynamic and time-sensitive, and they frequently exhibit small-world properties. [13] incorporate these distributional patterns as prior knowledge in evolutionary models to better capture network changes over time.

To conclude, communication networks often assume tree topologies and social networks are typically modeled as small-world networks. These assumptions do not hold for DFL network topologies. Moreover, common inference tools and metrics, such as traceroute, packet loss rate, or user interaction data, are not readily applicable in DFL contexts, creating the need for specialized inference metrics and strategies tailored to the particularities of DFL systems.

3 Overlay Topology in DFL

DFL is an ML paradigm that facilitates collaborative model training across multiple nodes without a central server. In DFL, each node updates its local model parameters using its private data and exchanges them with its directly connected neighbors. As the overlay topology in DFL significantly affects model convergence, this work presents the topology as an undirected graph $G = (V, E)$, where V is the set of nodes and E represents the edges indicating direct communication links between nodes. The adjacency matrix A encodes the graph structure, where:

$$A_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected,} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The degree matrix D is a diagonal matrix where $D_{ii} = \text{degree}(i) + 1$. The aggregation process is governed by the normalized aggregation matrix:

$$P = D^{-1}(A + I), \quad (2)$$

where I is the identity matrix. This formulation ensures P is row-stochastic ($\sum_j P_{ij} = 1$), enabling consistent scaling during aggregation.

DFL proceeds iteratively over T rounds, comprising local training and model aggregation steps. Let $M_t = [\theta_1^{(t)}, \theta_2^{(t)}, \dots, \theta_N^{(t)}]$ represent the models of all nodes at round t , where $\theta_i^{(t)}$ denotes the parameters of node i .

During the training stage, each node updates its model using its local dataset:

$$M_t = \tilde{M}_{t-1} + \delta_t, \quad (3)$$

where δ_t represents local updates, and \tilde{M}_{t-1} is the aggregated model from round $t - 1$.

In the aggregation stage, nodes aggregate models from neighbors using the aggregation matrix P :

$$\tilde{M}_t = PM_{t-1}. \quad (4)$$

The model parameters after T rounds can be expressed as:

$$M_T = P^T M_0 + \sum_{t=1}^T P^{T-t} \delta_t. \quad (5)$$

Assuming G is connected (i.e., no isolated nodes), the spectral radius of P satisfies $\rho(P) < 1$. Additionally, under bounded local updates (e.g., using SGD or Adam), there exists $C > 0$ such that $\|\delta_i^{(t)}\| \leq C$ for all t and i . As T increases, the influence of the initial model diminishes:

$$\|P^T M_0\| \leq \|P^T\| \|M_0\| \rightarrow 0 \quad \text{as } T \rightarrow \infty. \quad (6)$$

The cumulative effect of local updates is:

$$\sum_{t=1}^T P^{T-t} \delta_t, \quad (7)$$

where $\|P^{T-t} \delta_t\|$ is bounded by $\|P^{T-t}\| \|\delta_t\|$. Since $\|P^{T-t}\| \rightarrow 0$ as $T \rightarrow \infty$, the impact of individual updates diminishes, and the series converges due to $\rho(P) < 1$.

Thus, it can be seen that the convergence behavior of DFL is influenced by the topology captured by P . Initially, the aggregation dynamics (powers of P) dominate, reducing the influence of M_0 . Over time, local updates drive the learning process, modulated by the decaying powers of P . This interplay underscores the importance of overlay topology in determining convergence efficiency.

Given the critical role of overlay topology, it is essential to investigate potential information leakage through model updates and develop effective methods to protect sensitive topology information in DFL.

4 Problem Statement

Theoretical analysis of the DFL learning process reveals that overlay topology is crucial in determining system performance, robustness, and privacy. This section introduces the problem of topology inference and categorizes potential attack types based on the attacker's knowledge and capabilities.

Consider a DFL system represented by an undirected graph $G = (V, E)$, where V is the set of participating nodes, and E is the set of edges representing direct communication links between nodes. The adjacency matrix A of size $|V| \times |V|$ encodes the network topology, as defined in Equation (1).

An attacker aims to infer a predicted adjacency matrix A' that approximates the ground truth A as closely as possible, thereby revealing sensitive topology information.

The effectiveness of topology inference attacks depends on the attacker's knowledge and capabilities. The following assumptions are made. **(i) Internal Adversary:** The attacker is an internal participant in the DFL system and can identify a subset of nodes. **(ii) Decoupled Information:** Node-level information (e.g., models and datasets) and network-level information (e.g., connections) are treated as separate assets. For instance, an attacker may know a node's model but not its connections.

Let $V' \subset V$ represent a set of nodes known to the attacker and $E' \subset E$ represent the subset of edges known to the attacker, where $\emptyset \subsetneq E' \subsetneq E$ to represent partial knowledge. The attacker's goal is to infer the entire edge set E based on partial knowledge of V' and E' .

For each node $i \in V$, the local model is represented as M_i , and the local dataset is represented as D_i . This paper assumes that models and data within a node are distinct assets. Thus, the attacker's capabilities are defined as follows:

- **Model Knowledge:** The attacker can access models of known nodes V' , forming the set: $M' = \{M_i \mid i \in V'\}$.
- **Dataset Knowledge:** The attacker may not have access to the datasets D_i of these nodes. The datasets known to the attacker are denoted by: $D' \subseteq \{D_i \mid i \in V'\}$ or $D' = \emptyset$.

Based on the attacker's knowledge and capabilities, the topology inference attack is classified into five scenarios:

- **Scenario 1:** The attacker knows all node models and datasets, but only partial edge information ($M' = M, D' = D, \emptyset \subsetneq E' \subsetneq E$).
- **Scenario 2:** The attacker knows all node models, but none of the datasets, and only partial edge information ($M' = M, D' = \emptyset, \emptyset \subsetneq E' \subsetneq E$).
- **Scenario 3:** The attacker knows all node models and datasets, but does not have edge information ($M' = M, D' = D, E' = \emptyset$).
- **Scenario 4:** The attacker knows all node models, but neither datasets nor edge information ($M' = M, D' = \emptyset, E' = \emptyset$).
- **Scenario 5:** The attacker knows partial models, but neither datasets nor edge information ($\emptyset \subsetneq M' \subsetneq M, D' = \emptyset, E' = \emptyset$).

For Scenario 1, since the attacker can control all nodes and knows partial information about the edges, it can be classified as a white-box attack. Scenarios 2, 3, and 4 can be categorized as gray-box attacks. For attack scenario 5, no information is available regarding the total number of nodes or edges, making it difficult to fully reconstruct the network topology.

5 Topology Inference Attack

This section first explores the metrics that attackers can exploit to carry out topology inference attacks, and then designs attack strategies for various attack scenarios.

5.1 Attack Metrics

In a DFL system, attackers may leverage the local data and local models available at each known node. Additionally, DFL's decentralized communication mechanism allows attackers to access models transmitted from neighboring nodes. Thus, this paper proposes six attack metrics that attackers can exploit:

- **Relative Loss.** This metric evaluates how well a model trained on one node's data generalizes to another node's dataset. Consider a model f_i trained on node i 's dataset D_i , and let D_j be the dataset of another node j . The relative loss $\text{Relative Loss}_{i,j}$ measures the performance of f_i on D_j . A lower relative loss indicates better transferability and generalization across nodes:

$$\text{Relative Loss}_{i,j} = \mathcal{L}(f_i, D_j) \quad (8)$$

where $\mathcal{L}(f, D)$ is the loss function used by model f on dataset D .

- **Relative Entropy.** This metric measures the uncertainty of f_i 's predictions on D_j . While relative loss focuses on correctness, relative entropy focuses on confidence. A model may be confident (low entropy) but still perform poorly if it is incorrect.

$$\text{Relative Entropy}_{i,j} = -\frac{1}{|D_j|} \sum_{x \in D_j} \sum_k y_k(x) \log(f_{i,k}(x)) \quad (9)$$

where $y_k(x)$ is the true label distribution for sample x and $f_{i,k}(x)$ is the predicted probability of class k under model f_i .

- **Relative Sensitivity.** This metric quantifies the sensitivity of model f_i 's predictions on dataset D_j using the Jacobian norm. Relative sensitivity could capture the model's susceptibility to small input perturbation.

$$\text{Relative Sensitivity}_{i,j} = \frac{1}{|D_j|} \sum_{x \in D_j} \left\| \frac{\partial f_i(x)}{\partial x} \right\|_F \quad (10)$$

where $\frac{\partial f_i(x)}{\partial x}$ is the Jacobian of f_i 's outputs with respect to input x , and $\|\cdot\|_F$ denotes the Frobenius norm.

- **Cosine Similarity.** This metric compares the direction of two parameter vectors \mathbf{a} and \mathbf{b} derived from different models. If both vectors point in similar directions, their cosine similarity is high. In DFL, a high cosine similarity suggests that the two nodes frequently aggregate their parameters, resulting in more closely aligned models:

$$\text{Cosine Similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \quad (11)$$

where $\mathbf{a} \cdot \mathbf{b}$ is the dot product, and $\|\mathbf{a}\|, \|\mathbf{b}\|$ are the L2 norms of the vectors \mathbf{a} and \mathbf{b} , respectively.

- **Euclidean Similarity.** It quantifies how similar two vectors are based on the Euclidean distance between them. A higher Euclidean similarity indicates that the two models are numerically closer, while a smaller distance suggests they differ significantly:

$$\text{Euclidean Similarity}(\mathbf{a}, \mathbf{b}) = \frac{1}{1 + \sqrt{\sum_{i=1}^n (a_i - b_i)^2}} \quad (12)$$

where a_i and b_i are the i -th elements of \mathbf{a} and \mathbf{b} .

- **Curvature Divergence.** This metric quantifies model divergence by analyzing implicit second-order information from consecutive parameter updates, reflecting curvature characteristics without explicit data dependence. Specifically, it compares parameter update vectors to capture similarity in model evolution indirectly:

$$\text{Curvature Divergence}(\Delta \mathbf{a}, \Delta \mathbf{b}) = \frac{\|\Delta \mathbf{a} - \Delta \mathbf{b}\|_2}{\frac{1}{2}(\|\Delta \mathbf{a}\|_2 + \|\Delta \mathbf{b}\|_2)} \quad (13)$$

where $\Delta \mathbf{a} = \mathbf{a}^t - \mathbf{a}^{t-1}$ and $\Delta \mathbf{b} = \mathbf{b}^t - \mathbf{b}^{t-1}$ denote parameter updates between rounds $t-1$ and t . A lower value indicates lower divergence (i.e., more similar updates).

Relative metrics, including relative loss, entropy, and sensitivity, require access to both local models and local datasets, thus making them suitable for **attack scenarios 1 and 3**. While similarity metrics, *i.e.* cosine, Euclidean, and curvature similarity, depend only on the models and do not require dataset access, they are appropriate for **attack scenarios 2 and 4**.

5.2 Attack Strategies

In a topology inference attack, the attacker's objective is to construct a predicted adjacency matrix A' that closely approximates the ground-truth adjacency matrix A .

5.2.1 Supervised Attack Strategy.

In **attack scenarios 1 and 2**, where the attacker has access to partial ground-truth edge labels, this work introduces *EDGEPRE*, a supervised edge classification framework for topology reconstruction. As shown in Algorithm 1, the model learns to infer the full adjacency matrix from node-level behavioral features and a small set of labeled node pairs.

Model Architecture. Given N nodes and their behavioral metrics $X \in \mathbb{R}^{N \times d}$, *EDGEPRE* directly uses an MLP-based edge classifier without any message passing. For each labeled pair (i, j) , the concatenated feature pair $[x_i \| x_j]$ is passed to a multilayer perceptron (MLP) decoder that outputs the predicted edge probability:

$$\hat{y}_{ij} = \sigma(f_{\text{dec}}([x_i \| x_j])).$$

Optionally, feature interaction terms such as $x_i * x_j$ and $|x_i - x_j|$ can be included to enhance the representation.

Training Objective. The model is optimized using binary cross-entropy over the labeled edge set:

$$\mathcal{L} = \sum_{(i,j,y_{ij}) \in \mathcal{P}} \text{BCE}(\hat{y}_{ij}, y_{ij}).$$

Adjacency Reconstruction. After training, *EDGEPRE* generalizes to all node pairs and produces a reconstructed soft adjacency matrix $A' \in [0, 1]^{N \times N}$, where each entry is computed via:

$$A'_{ij} = \sigma(f_{\text{dec}}([x_i \| x_j])).$$

Algorithm 1 EDGEPRE (MLP-BASED)

Require: Node features $X \in \mathbb{R}^{N \times d}$, labeled edge set $\mathcal{P}_{\text{train}} = \{(i, j, y_{ij})\}$, number of epochs T
Ensure: Reconstructed adjacency matrix $A' \in [0, 1]^{N \times N}$

- 1: Initialize MLP decoder f_{dec}
- 2: **for** $t = 1$ to T **do**
- 3: **for** each labeled pair $(i, j, y_{ij}) \in \mathcal{P}_{\text{train}}$ **do**
- 4: $h_{ij} \leftarrow [x_i \| x_j]$
- 5: $\hat{y}_{ij} \leftarrow \sigma(f_{\text{dec}}(h_{ij}))$
- 6: Compute loss: $\mathcal{L} \leftarrow \sum_{(i,j)} \text{BCE}(\hat{y}_{ij}, y_{ij})$
- 7: Update decoder parameters via gradient descent
- 8: **Inference phase**
- 9: **for** all (i, j) where $1 \leq i, j \leq N$ **do**
- 10: $h_{ij} \leftarrow [x_i \| x_j]$
- 11: $A'_{ij} \leftarrow \sigma(f_{\text{dec}}(h_{ij}))$
- 12: **return** A'

5.2.2 Unsupervised Attack Strategy.

For **attack scenarios 3 and 4**, where no edge labels are available, this work proposes *INFERGAT*, an unsupervised structure inference method based on a GAT encoder-decoder framework. The model predicts the latent adjacency purely from node-level behavioral metrics without requiring ground-truth edges.

Model Architecture. The input $X \in \mathbb{R}^{N \times d}$ contains attack-related features such as similarity or entropy. A GAT encoder outputs embeddings $Z = f_{\text{enc}}(X)$, and a symmetric MLP decoder estimates pairwise edge strength via $A'_{ij} = \sigma(\text{MLP}([z_i \| z_j]))$.

Training Objective. The model is trained to reconstruct the behavioral structure with mean squared error:

$$\mathcal{L} = \sum_{i,j} \|A'_{ij} - X_{ij}\|^2 \quad (14)$$

The final output is the predicted adjacency matrix $A' \in [0, 1]^{N \times N}$.

Algorithm 2 INFERGAT

Require: Attack metrics $X \in \mathbb{R}^{N \times d}$, learning rate η , epochs T
Ensure: Predicted adjacency matrix $A' \in [0, 1]^{N \times N}$

- 1: Initialize GAT encoder f_{enc} and decoder f_{dec} with parameters θ
- 2: **for** $t = 1$ to T **do**
- 3: $Z \leftarrow f_{\text{enc}}(X)$ ▷ Node embeddings via multi-head GAT
- 4: $A'_{ij} \leftarrow f_{\text{dec}}(z_i, z_j)$, $\forall i, j$ ▷ Decoder predicts edge weights
- 5: $A' \leftarrow \frac{1}{2}(A' + A'^\top)$ ▷ Symmetrize adjacency
- 6: $\mathcal{L} \leftarrow \sum_{i,j} \|A'_{ij} - X_{ij}\|^2$ ▷ Reconstruction loss
- 7: $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}$ ▷ Gradient descent update
- 8: **return** A'

EDGEPRE and *INFERGAT* represent two attack strategies for topology inference in DFL. *EDGEPRE* leverages partial edge supervision to perform explicit edge classification, while *INFERGAT* relies solely on behavioral metrics to reconstruct structural patterns in a self-supervised manner. Together, they cover a broad range of adversarial capabilities and reflect realistic threat models across different attack scenarios.

6 Evaluation

This work empirically validates the effectiveness of the proposed topology inference attack. First, it identifies which metrics yield stronger attack performance. Building on these results, it then thoroughly evaluates the proposed attack strategies under various conditions, including different datasets, topologies, node counts, and data preprocessing methods.

6.1 Experimental Setups

This section describes the configurations employed in the experiments.

A. Dataset and Model: Experiments are conducted on: MNIST, FMNIST, CIFAR10, SVHN, ImageNet10, and Malware, which are widely used in FL and inference attack benchmarks.

- **MNIST [21]:** Contains 60,000 grayscale training images and 10,000 test images of handwritten digits (0–9) at 28×28 resolution. A two-hidden-layer MLP (256 and 128 neurons) trained with Adam (lr=1e-3) is used.
- **FMNIST [31]:** Similar size and format to MNIST, but with 10 categories of clothing items. A CNN with two convolutional layers (32 and 64 filters, kernel size 3×3) is employed.
- **CIFAR10 [20]:** Contains 60,000 32×32 RGB images from 10 categories. The training set is only treated with basic image normalization. The image pixel values are normalized to keep the original image content unchanged. A MobileNet [26] is used for training.
- **SVHN [23]:** Composes of over 600,000 32×32 RGB images of house numbers collected from Google Street View. Each image contains a centered digit from 0 to 9. A ResNet9 [16] is adopted as the training model.
- **ImageNet10:** A curated subset of the ImageNet [6], containing 10 classes selected from the original ImageNet hierarchy. This version [27] includes approximately 15,000 images with varied object categories and balanced class distributions. A lightweight PoolFormer-S12 [33] model is used for training.
- **Malware [8]:** The Malware dataset comprises 342,106 tabular records with 31 extracted features, collected from eight IoT crowdsensing devices infected by eight types of malware. Each

record is labeled as either the corresponding malware type or normal behavior. A MLP with two hidden layers (64 and 32 neurons) is employed.

B. DFL Topology Setting: All experiments are conducted on DFL systems comprising two categories of topologies: synthetic and real-world. The synthetic topologies include ring, star, and three Erdős-Rényi (ER) graphs with connection probabilities $p = 0.3, 0.5$, and 0.7 . The real-world topologies consist of 27 network topologies collected from the SNDlib [24] and DEFO [15] repositories, which reflect realistic communication and infrastructure networks. Detailed descriptions and properties of these topologies are provided in the supplementary material.

C. Federation Setting: All experiments employ the following DFL configurations:

- **Number of Nodes:** For synthetic topologies, the number of nodes is set to 10, 20, and 30. For real-world topologies, the number of nodes ranges from 12 to 87, depending on the specific topology.
- **Total Rounds:** The total number of communication rounds is determined by the rule $\text{rounds} = \text{number of nodes}$. The mechanism ensures efficient inter-node communication, thereby enabling consistent scalability of training time as the network size increases.
- **Local Epochs:** Two scenarios (3 and 10 local epochs) are considered to evaluate how varying levels of local overfitting affect the attack’s effectiveness.
- **Data Distribution:** Both Independent and Identically Distributed (IID) and non-IID $\alpha = 0.1$ data distributions are considered to evaluate the impact of data heterogeneity.

D. Attack Models: To evaluate the effectiveness of the proposed methods, *EDGEPRE* is compared against classical supervised classifiers, including Logistic Regression, Support Vector Machine (SVM), and Random Forest (RF). Additionally, graph-specific supervised baselines are also used, including: *SEAL* [34] and *GraphSAGE* (edge classification setting) [14].

For the unsupervised attack strategy, *INFERGAT* is evaluated against standard clustering-based methods, including K-Means, Gaussian Mixture Model (GMM), and Spectral Clustering. This experiment further adds modern graph-reconstruction and self-supervised baselines: *GRACE* [35] and *BGRL* [29].

E. Evaluation Metric: Since the topology inference attack is formulated as a binary classification problem, evaluation uses the **F1-Score** and **AUC-ROC** metrics.

6.2 Selection of Attack Metrics

Selecting appropriate attack metrics is critical to the success of inference attacks. An effective attack metric should distinguish between connected and non-connected nodes as clearly as possible. To this end, the first part of this experiment examines the effectiveness of the six proposed attack metrics by analyzing their respective distributions.

Figure 1 illustrates the distributions of these metrics for a 20-node star topology DFL system trained on the FMNIST dataset, comparing directly connected nodes (edge group) with those not (non-edge group). To facilitate direct comparison, values have been normalized.

The results indicate that the relative loss for connected nodes is notably lower than that for non-connected nodes, with minimal overlap between the respective distributions. By contrast, the relative entropy and sensitivity metrics exhibit substantially greater overlap. Since

Scenario	Algorithm	CIFAR10		FMNIST		ImageNet10		Malware		MNIST		SVHN	
		F1	AUC										
SC1	Logistic	0.806 ± 0.129	0.519 ± 0.064	0.804 ± 0.126	0.500 ± 0.000	0.804 ± 0.126	0.500 ± 0.000	0.804 ± 0.126	0.500 ± 0.000	0.800 ± 0.125	0.500 ± 0.000	0.804 ± 0.126	0.500 ± 0.000
	RF	0.795 ± 0.132	0.643 ± 0.127	0.786 ± 0.142	0.653 ± 0.128	0.833 ± 0.107	0.695 ± 0.132	0.714 ± 0.144	0.503 ± 0.050	0.765 ± 0.135	0.600 ± 0.102	0.788 ± 0.114	0.633 ± 0.116
	SVM	0.840 ± 0.102	0.609 ± 0.153	0.818 ± 0.115	0.561 ± 0.122	0.808 ± 0.118	0.519 ± 0.066	0.800 ± 0.137	0.499 ± 0.025	0.801 ± 0.131	0.519 ± 0.067	0.824 ± 0.108	0.566 ± 0.130
	GraphSage	0.740 ± 0.077	0.607 ± 0.253	0.699 ± 0.058	0.474 ± 0.188	0.701 ± 0.045	0.507 ± 0.209	0.686 ± 0.043	0.423 ± 0.186	0.686 ± 0.039	0.460 ± 0.185	0.712 ± 0.056	0.526 ± 0.200
	SEAL	0.701 ± 0.057	0.560 ± 0.142	0.697 ± 0.052	0.556 ± 0.152	0.699 ± 0.057	0.557 ± 0.156	0.705 ± 0.052	0.577 ± 0.144	0.701 ± 0.050	0.584 ± 0.129	0.697 ± 0.056	0.548 ± 0.146
SC2	EDGEPRE	0.847 ± 0.101	0.831 ± 0.136	0.873 ± 0.096	0.857 ± 0.135	0.855 ± 0.101	0.826 ± 0.181	0.858 ± 0.102	0.834 ± 0.192	0.862 ± 0.095	0.839 ± 0.144	0.886 ± 0.091	0.871 ± 0.130
	Logistic	0.811 ± 0.128	0.543 ± 0.101	0.804 ± 0.126	0.500 ± 0.000	0.801 ± 0.127	0.500 ± 0.001	0.804 ± 0.126	0.500 ± 0.000	0.800 ± 0.125	0.500 ± 0.000	0.804 ± 0.126	0.500 ± 0.000
	RF	0.787 ± 0.134	0.629 ± 0.132	0.762 ± 0.135	0.568 ± 0.107	0.784 ± 0.122	0.579 ± 0.096	0.722 ± 0.139	0.505 ± 0.048	0.746 ± 0.140	0.562 ± 0.082	0.765 ± 0.129	0.598 ± 0.102
	SVM	0.836 ± 0.108	0.604 ± 0.151	0.817 ± 0.117	0.559 ± 0.119	0.810 ± 0.116	0.520 ± 0.067	0.800 ± 0.137	0.499 ± 0.025	0.800 ± 0.132	0.517 ± 0.066	0.821 ± 0.110	0.560 ± 0.128
	GraphSage	0.697 ± 0.050	0.488 ± 0.172	0.700 ± 0.069	0.510 ± 0.152	0.699 ± 0.050	0.501 ± 0.135	0.700 ± 0.056	0.473 ± 0.157	0.698 ± 0.056	0.498 ± 0.139	0.699 ± 0.060	0.490 ± 0.152
SC3	SEAL	0.699 ± 0.042	0.558 ± 0.137	0.704 ± 0.052	0.582 ± 0.108	0.700 ± 0.046	0.589 ± 0.105	0.700 ± 0.050	0.561 ± 0.145	0.701 ± 0.049	0.570 ± 0.133	0.706 ± 0.051	0.579 ± 0.130
	EDGEPRE	0.786 ± 0.093	0.721 ± 0.175	0.825 ± 0.108	0.791 ± 0.168	0.811 ± 0.091	0.772 ± 0.153	0.822 ± 0.112	0.778 ± 0.188	0.827 ± 0.108	0.797 ± 0.161	0.826 ± 0.109	0.784 ± 0.167
	BGRL	0.384 ± 0.344	0.500 ± 0.000	0.384 ± 0.344	0.500 ± 0.000	0.384 ± 0.344	0.500 ± 0.000	0.384 ± 0.344	0.500 ± 0.000	0.433 ± 0.349	0.500 ± 0.000	0.384 ± 0.344	0.500 ± 0.000
	GMM	0.743 ± 0.252	0.674 ± 0.081	0.742 ± 0.201	0.783 ± 0.129	0.668 ± 0.227	0.731 ± 0.161	0.795 ± 0.107	0.748 ± 0.136	0.864 ± 0.150	0.890 ± 0.156	0.852 ± 0.186	0.869 ± 0.126
	GRACE	0.866 ± 0.000	0.426 ± 0.207	0.534 ± 0.469	0.446 ± 0.181	0.866 ± 0.000	0.396 ± 0.207	0.500 ± 0.337	0.441 ± 0.189	0.534 ± 0.469	0.438 ± 0.186	0.866 ± 0.000	0.429 ± 0.210
SC4	Kmeans	0.793 ± 0.090	0.636 ± 0.087	0.741 ± 0.187	0.891 ± 0.134	0.792 ± 0.223	0.774 ± 0.181	0.779 ± 0.101	0.764 ± 0.164	0.754 ± 0.178	0.820 ± 0.152	0.907 ± 0.109	0.934 ± 0.093
	Spectral	0.727 ± 0.168	0.592 ± 0.137	0.710 ± 0.268	0.594 ± 0.189	0.684 ± 0.314	0.709 ± 0.172	0.875 ± 0.111	0.721 ± 0.207	0.677 ± 0.231	0.594 ± 0.189	0.760 ± 0.148	0.569 ± 0.146
	INFERGAT	0.893 ± 0.103	0.876 ± 0.161	0.850 ± 0.116	0.789 ± 0.147	0.765 ± 0.158	0.813 ± 0.157	0.664 ± 0.154	0.793 ± 0.158	0.657 ± 0.199	0.860 ± 0.186	0.769 ± 0.184	0.692 ± 0.391

Table 1. F1 and AUC (mean ± std) across datasets. Best per scenario/dataset in **bold**.

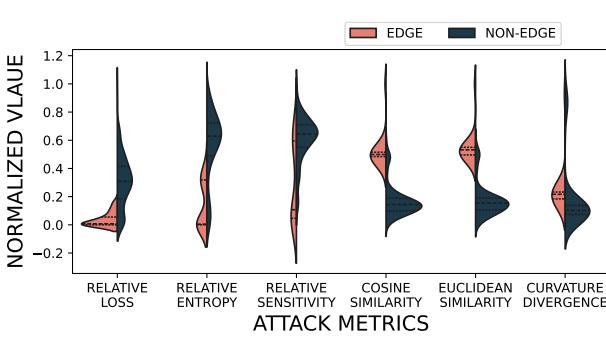


Figure 1. Normalized distributions of attack metrics in a 20-node star-topology DFL system trained on the FMNIST dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection.

computing the relative loss requires access to local models and data, it has been employed in attack scenarios 1 and 3.

Regarding similarity-based metrics, the cosine similarity between connected models is distinctly higher than that between non-connected models, displaying no overlap between the distributions. On the other hand, the curvature divergence and Euclidean similarity show a considerably more significant degree of overlap. Thus, the cosine similarity has been adopted as the attack metric in scenarios where only local models are available (attack scenarios 2 and 4). The complete set of experimental results for all datasets and topologies is in the Appendix.

6.3 Attack Performance

After establishing the attack metrics, the proposed topology inference attacks are evaluated on six datasets under four scenarios. As shown in Table 1, *EDGEPRE* dominates in SC1–SC2, while *INFERGAT* ranks best in SC3–SC4 across all datasets.

Performance decreases as the attacker’s prior knowledge is reduced from SC1 to SC4. In SC1, *EDGEPRE* attains F1 of at least ≈0.85 on every dataset (with high AUC as well). Moving to SC2

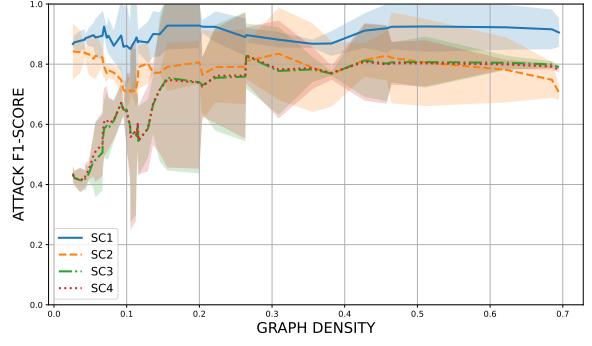


Figure 2. Impact of topology density on the F1-scores achieved by each attack scenario on the CIFAR10 dataset.

yields only a modest drop (typically a few points), indicating that model parameters alone already encode most of the topology signal. The larger degradation appears when supervision is removed: in SC3, *INFERGAT* remains strongest but F1 becomes dataset-dependent (e.g., ≈0.89 on CIFAR10 vs. ≈0.66–0.77 on MNIST and Malware). Under the weakest prior (SC4), *INFERGAT* still outperforms other unsupervised baselines, achieving F1 around 0.60–0.67 with AUC ≈0.76–0.80. Overall, these trends confirm that stronger prior/model knowledge markedly improves topology inference, whereas limited priors and no labels (SC3 and SC4) lead to the main loss in accuracy.

A. Effect of Topology Density. To investigate the influence of topological characteristics on topology inference attacks, Figure 2 reports the F1-Scores for four attack scenarios on the CIFAR10 dataset. Topology density, defined as $\frac{2E}{N(N-1)}$, reflects overall connectivity.

Overall, the four scenarios exhibit relatively stable performance across the full range of densities. For SC3 and SC4, which represent unsupervised attacks, denser topologies provide richer edge information that supports the recovery of the topology, leading to gradually improved performance as connectivity increases. In contrast, SC1 and SC2 achieve comparatively high scores across all densities, reflecting their ability to exploit both sparse and dense settings. The shaded areas further indicate that variability is higher at very low densities, where limited connections introduce uncertainty, while the results become stable once sufficient connectivity is reached.

B. Effect of Network Size. To evaluate the scalability of the inference algorithms under different attack scenarios, Figure 3 illustrates the variation in F1-Scores as the network size increases. SC1 and SC2 remain relatively stable across different numbers of nodes, indicating robustness to changes in network size. In contrast, SC3 and SC4 exhibit a clear decline as the number of nodes grows. This behavior is explained by the fact that larger networks typically correspond to lower densities, which reduce the amount of effective edge information available for unsupervised inference. Consequently, the performance of SC3 and SC4 degrades with increasing network size, whereas the supervised scenarios (SC1 and SC2) maintain more stable effectiveness.

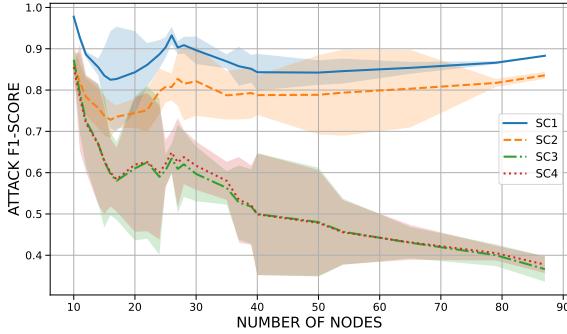


Figure 3. Impact of network size on the F1-scores achieved by each attack scenario on the CIFAR10 dataset.

7 Mitigation

The effectiveness of the proposed topology inference attacks indicates that topology in DFL systems can be discerned solely by analyzing model behaviors. This observation highlights the security and privacy concerns of DFL systems. This section proposes defense strategies to mitigate these vulnerabilities within DFL environments.

7.1 Reduce Overfitting

As illustrated in Figure 4, each bar represents a specific training configuration: gray bars denote IID data distributions without augmentation and blue bars correspond to non-IID data distributions (Dirichlet $\alpha = 0.1$).

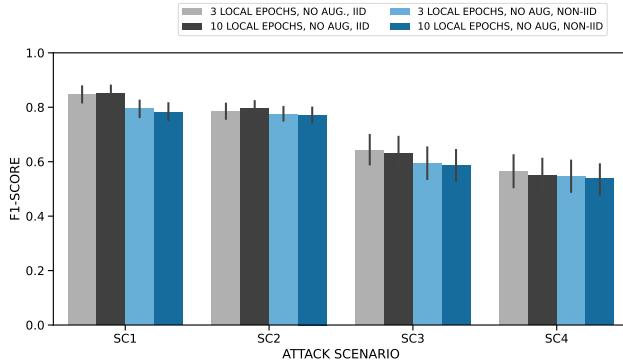


Figure 4. Impact of local training epochs and data heterogeneity on the F1-Score of topology inference attacks on the CIFAR10 dataset.

Consistent with findings in prior inference attack literature, the success of topology inference attacks is closely tied to the degree of overfitting in the target model. To validate this, experiments were conducted on the CIFAR10 dataset under four attack scenarios. By increasing the number of local training epochs from 3 to 10, the level of overfitting was elevated. Results show that, across all examined topologies, this change consistently improves the F1-Scores of attacks. This observation confirms that overfitting exacerbates the risk of topological information leakage in DFL systems.

7.2 Data Heterogeneity

Furthermore, data distribution among clients plays a critical role in attack effectiveness. Non-IID data were simulated using a Dirichlet distribution with $\alpha = 0.1$ to assess this. The results, represented by the blue bars, indicate a noticeable decline in attack performance across all scenarios, with the effect being most prominent in scenarios 3 and 4. This suggests that increased data heterogeneity introduces higher model divergence, which in turn hampers the inference model's ability to exploit overfitted representations for edge prediction.

7.3 Differential Privacy

Differential Privacy (DP) is a common mitigation for privacy in FL [2]. DP is integrated into DFL parameter sharing and its effect on topology inference is assessed. On CIFAR10 (ER_0.5) under SC1, attack F1 drops from 0.889 \rightarrow 0.687 (10 nodes) and 0.824 \rightarrow 0.637 (20 nodes), approximately 23% reductions, indicating effective mitigation.

Overall, these findings imply that reducing overfitting through fewer local training epochs and increasing data heterogeneity are promising directions for mitigating topology inference risks in DFL.

8 Conclusion and Future Work

This work introduces a novel topology inference attack against DFL, exposing critical vulnerabilities related to privacy and information leakage. By analyzing local models, attackers can accurately infer the overlay topology, one of DFL's most sensitive assets, highlighting the system's susceptibility to privacy breaches. The study explores various attack scenarios and develops tailored metrics, models, and algorithms, with experiments confirming the feasibility and effectiveness of these attacks. Furthermore, network size, density, model overfitting, and data heterogeneity significantly influence attack success. Mitigation strategies, such as increasing data heterogeneity, can enhance model generalization and reduce the risk of data leakage.

The present work focuses on attack scenarios 1 through 4 and does not yet address the most challenging attack scenario 5. Future research plans include exploring feasible strategies for Scenario 5 and extending the evaluations to a wider range of datasets and topologies. Besides, more attack metrics and strategies will be developed to improve attack effectiveness.

Acknowledgment

This work has been partially supported by (a) the Swiss Federal Office for Defense Procurement (armasuisse) with the CyberDFL project (CYD-C-2020003) and (b) the University of Zürich UZH.

References

[1] E. T. M. Beltrán, Á. L. P. Gómez, C. Feng, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán. Fedstellar: A platform for decentralized federated learning. *Expert Systems with Applications*, 242:122861, 2024.

[2] E. T. Beltrán, M. Q. Pérez, P. M. S. Sánchez, S. L. Bernal, G. Bovet, M. G. Pérez, G. M. Pérez, and A. H. Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys and Tutorials*, 25(4):2983–3013, 2023. doi: 10.1109/COMST.2023.3315746.

[3] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: Recent developments, 2004.

[4] A. Coates, A. O. Hero III, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal processing magazine*, 19(3):47–65, 2002.

[5] S. Dayal, D. Alhadidi, A. Abbasi Tadi, and N. Mohammed. Comparative analysis of membership inference attacks in federated learning. In *Proceedings of the 27th International Database Engineered Applications Symposium*, pages 185–192, 2023.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] C. Feng, A. H. Celdrán, J. Von der Assen, E. T. M. Beltrán, G. Bovet, and B. Stiller. Dart: A solution for decentralized federated learning model robustness analysis. *Array*, page 100360, 2024.

[8] C. Feng, A. H. Celdran, J. Han, H. Ren, X. Cheng, Z. Zeng, L. Krauter, G. Bovet, and B. Stiller. A crowdsensing intrusion detection dataset for decentralized federated learning models. *arXiv preprint arXiv:2507.13313*, 2025.

[9] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

[10] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.

[11] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):1–37, 2012.

[12] N. Z. Gong and B. Liu. Attribute inference attacks in online social networks. *ACM Transactions on Privacy and Security (TOPS)*, 21(1):1–30, 2018.

[13] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song. Evolution of social-attribute networks: measurements, modeling, and implications using google+. In *Proceedings of the 2012 internet measurement conference*, pages 131–144, 2012.

[14] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[15] R. Hartert, S. Vissicchio, P. Schaus, O. Bonaventure, C. Filsfils, T. Telkamp, and P. Francois. A declarative and expressive approach to control forwarding paths in carrier-grade networks. *ACM SIGCOMM computer communication review*, 45(4):15–28, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] T. Hou, Z. Qu, T. Wang, Z. Lu, and Y. Liu. Proto: Proactive topology obfuscation against adversarial network topology inference. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1598–1607. IEEE, 2020.

[18] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11):1–37, 2022.

[19] X. Jin, W.-P. K. Yiu, S.-H. G. Chan, and Y. Wang. Network topology inference based on end-to-end measurements. *IEEE Journal on Selected areas in Communications*, 24(12):2182–2195, 2006.

[20] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images, 2009.

[21] Y. LeCun and C. Cortes. MNIST handwritten digit database. <http://yann.lecun.com/exdb/mnist/>, 2010. Accessed: 2016-01-14.

[22] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[23] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[24] S. Orlowski, R. Wessely, M. Pióro, and A. Tomaszewski. Sndl1.0—survivable network design library. *Networks: An International Journal*, 55(3):276–286, 2010.

[25] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. URL <https://arxiv.org/abs/1801.04381>.

[27] L. Sha. Imagenet10 (kaggle version). <https://www.kaggle.com/datasets/liusha249/imagenet10>, 2022.

[28] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017. doi: 10.1109/SP.2017.41.

[29] S. Thakoor, C. Tallec, M. G. Azar, R. Munos, P. Veličković, and M. Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*, 2021.

[30] Y. Wang, Y.-X. Wang, and A. Singh. Graph connectivity in noisy sparse subspace clustering. In *Artificial Intelligence and Statistics*, pages 538–546. PMLR, 2016.

[31] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747 [cs, stat]*, Sept. 2017. doi: 10.48550/arXiv.1708.07747. Accessed: 2023-07-11.

[32] X. Yin, Y. Zhu, and J. Hu. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021.

[33] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[34] M. Zhang and Y. Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[35] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

A appendix

A.1 Topology Information

This work utilizes two categories of topologies: synthetic and real-world. The synthetic topologies include ring, star, and three Erdős-Rényi (ER) graphs with connection probabilities $p = 0.3, 0.5$, and 0.7 . The real-world topologies consist of 27 network topologies collected from the two public network repositories: SNDlib² and DEFO³, which reflect realistic communication and infrastructure networks.

To characterize the structure of different graphs, this work reports the following topological statistics:

- **Number of Nodes:** The total number of distinct vertices in the graph, denoted as N .
- **Number of Edges:** The total number of edges connecting node pairs, denoted as E .
- **Average Degree:** The mean number of edges connected to each node, calculated as $\frac{2E}{N}$.
- **Density:** The ratio of the number of actual edges to the maximum possible number of edges, given by $\frac{2E}{N(N-1)}$. This metric reflects the graph's overall connectivity or sparsity.

A.1.1 Synthetic Topology Characterize Statistics

To evaluate the robustness and generalizability of the proposed method under various topology conditions, this work considers a set of synthetic topologies ranging from sparse to dense:

- **STAR graphs** represent extremely sparse topologies, where most nodes are only connected to a single central hub. These graphs have low average degree and density, reflecting minimal connectivity.
- **RING graphs** are sparse structures with uniform and low node degrees. While slightly denser than STAR graphs, they remain sparse and exhibit limited path diversity.
- **ER graphs** with varying edge probabilities ($p = 0.3, 0.5$, and 0.7) capture a continuum of connectivity levels. As p increases, the average degree and edge density grows, gradually transitioning the graph from a sparse to a dense structure.

This selection enables controlled evaluation across different topological regimes, ensuring that performance is not biased toward a specific graph sparsity or structure. The characterize statistics of synthetic topologies are shown in Table 2

Table 2. Topological Statistics of Different Synthetic Topology

Topology	Number of Nodes	Number of Edges	Average Degree	Density
ER($p = 0.3$)	10	16	3.20	0.36
ER($p = 0.3$)	30	115	7.67	0.26
ER($p = 0.3$)	20	50	5.00	0.26
ER($p = 0.5$)	10	23	4.60	0.51
ER($p = 0.5$)	30	202	13.47	0.46
ER($p = 0.5$)	20	87	8.70	0.46
ER($p = 0.7$)	20	132	13.20	0.69
ER($p = 0.7$)	30	298	19.87	0.69
ER($p = 0.7$)	10	28	5.60	0.62
RING	10	10	2.00	0.22
RING	20	20	2.00	0.11
RING	30	30	2.00	0.07
STAR	10	9	1.80	0.20
STAR	20	19	1.90	0.10
STAR	30	29	1.93	0.07

A.1.2 Real-world Topology Characterize Statistics

To evaluate the generalizability and robustness of the proposed approach across diverse topological structures, experiments were conducted on a wide range of real-world network topologies. These networks vary significantly in size, ranging from small-scale graphs with only 10 nodes to large-scale structures with up to 87 nodes. The topologies are sourced from realistic communication infrastructures, including national research and education networks (NRENs) and Internet service provider (ISP) backbone networks, which reflect actual deployment scenarios.

A statistical overview of these networks is provided in Table 3. Compared to synthetic graphs, real-world networks exhibit higher structural variability in terms of node degree, connectivity, and scale. Despite this diversity, they are generally characterized by sparse connectivity, as indicated by low edge densities and moderate average degrees. This sparsity is a common property of real communication networks and introduces specific challenges to graph data.

² Available in: <https://sndlib.put.poznan.pl/home.action>

³ Available in: <https://sites.uclouvain.be/defo/>

Table 3. Topological statistics of real-world and synthetic networks

Topology	Number of Nodes	Number of Edges	Average Degree	Density
ABILENE	12	15	2.50	0.11
ATLANTA	15	22	2.93	0.10
COST266	37	57	3.08	0.04
DFN-GWIN	11	47	8.55	0.43
DI-YUAN	11	42	7.64	0.38
FRANCE	25	45	3.60	0.08
GEANT	22	72	6.55	0.16
GERMANY50	50	88	3.52	0.04
GIUL39	39	172	8.82	0.12
INDIA35	35	80	4.57	0.07
JANOS-US	26	84	6.46	0.13
JANOS-US-CA	39	122	6.26	0.08
NEWYORK	16	49	6.13	0.20
NOBEL-EU	28	41	2.93	0.05
NOBEL-GERMANY	17	26	3.06	0.10
NOBEL-US	14	21	3.00	0.12
NORWAY	27	51	3.78	0.07
PDH	11	34	6.18	0.31
PIORO40	40	89	4.45	0.06
POLSKA	12	18	3.00	0.14
RF1755	87	322	7.40	0.04
RF3967	79	294	7.44	0.05
SUN	27	102	7.56	0.15
SYNTH50	50	276	11.04	0.11
TA1	24	51	4.25	0.09
TA2	65	108	3.32	0.03
ZIB54	54	80	2.96	0.03

A.2 Attack Metrics Distribution

This experiment comprehensively demonstrates the distribution of attack metrics in the edge and non-edge groups across CIFAR10 (Figure 5), MNIST (Figure 6), FMNIST (Figure 7), SVHN (Figure 8), ImageNet10 (Figure 9), and Malware (Figure 10) datasets in various topologies. It can be observed that, in most cases, the distribution of relative loss is more concentrated than that of relative entropy and sensitivity. Additionally, relative loss exhibits less overlap between the edge group and the non-edge group. Therefore, this work selects relative loss as the attack metric when the attacker has access to both local data and the local model, *i.e.*, attack scenarios 1 and 3.

Regarding similarity-based metrics, the cosine similarity between connected models is distinctly higher than that between non-connected models, displaying no overlap between the distributions. On the other hand, the curvature divergence and Euclidean similarity show a considerably more significant degree of overlap. Thus, the cosine similarity has been adopted as the attack metric in scenarios where only local models are available (attack scenarios 2 and 4).

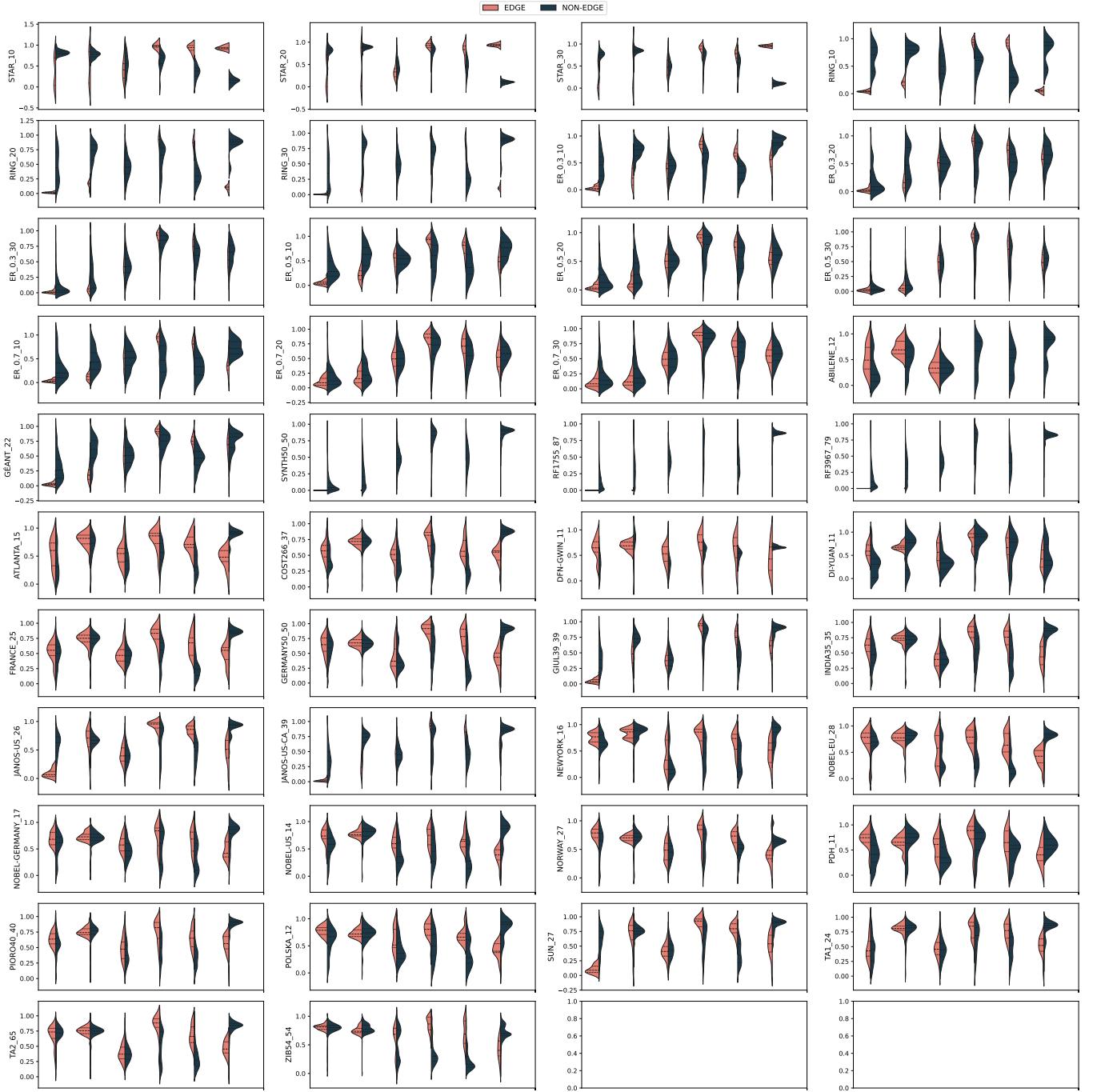


Figure 5. Normalized distributions of attack metrics in various topology DFL systems trained on the **CIFAR10** dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection. In each subfigure, the attack metrics are respectively: Relative Loss, Relative Entropy, Relative Sensitivity, Cosine Similarity, Euclidean Similarity, and Curvature Divergence.

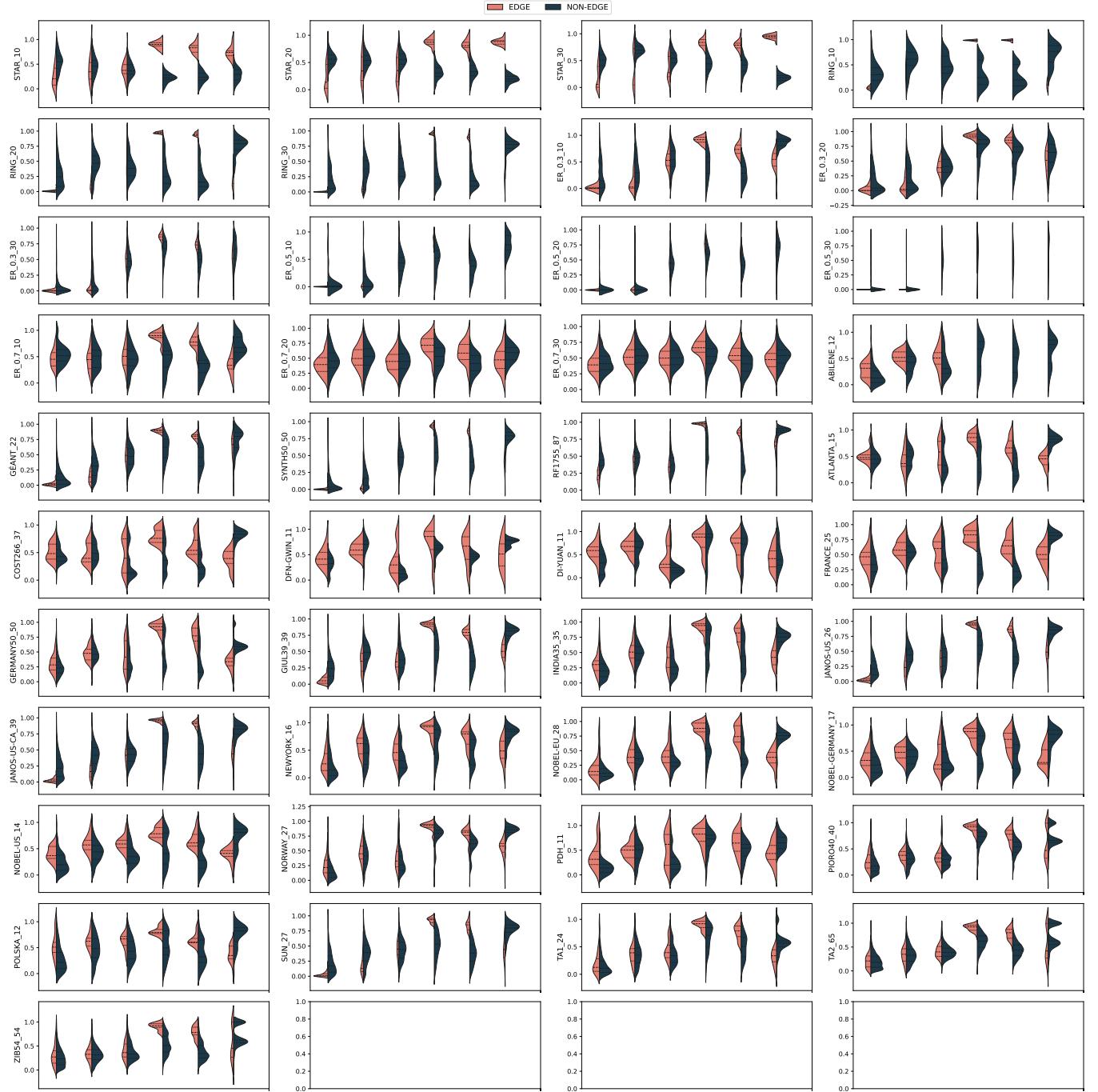


Figure 6. Normalized distributions of attack metrics in various topology DFL systems trained on the **MNIST** dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection. In each subfigure, the attack metrics are respectively: Relative Loss, Relative Entropy, Relative Sensitivity, Cosine Similarity, Euclidean Similarity, and Curvature Divergence.



Figure 7. Normalized distributions of attack metrics in various topology DFL systems trained on the **FMNIST** dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection. In each subfigure, the attack metrics are respectively: Relative Loss, Relative Entropy, Relative Sensitivity, Cosine Similarity, Euclidean Similarity, and Curvature Divergence.

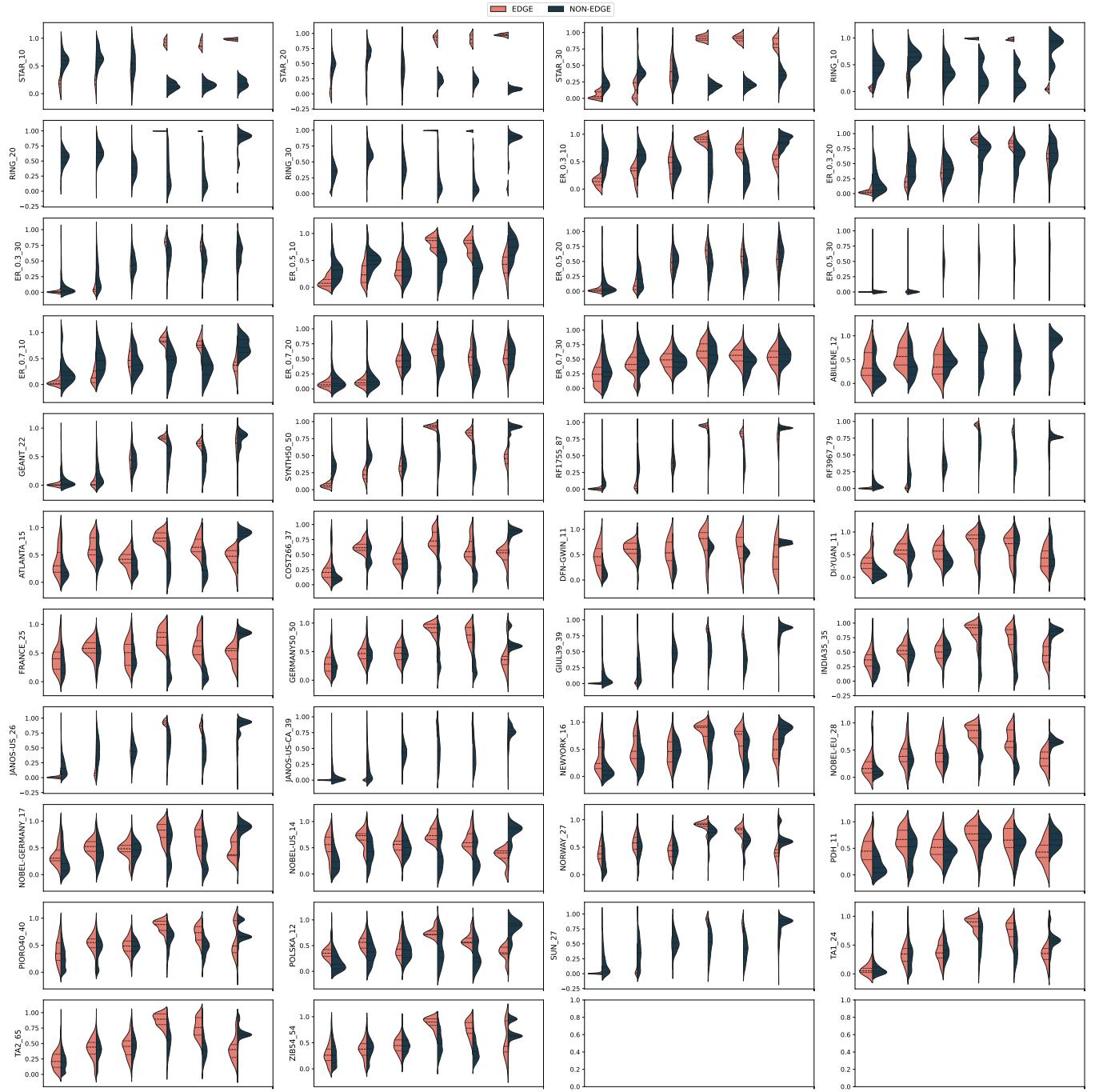


Figure 8. Normalized distributions of attack metrics in various topology DFL systems trained on the **SVHN** dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection. In each subfigure, the attack metrics are respectively: Relative Loss, Relative Entropy, Relative Sensitivity, Cosine Similarity, Euclidean Similarity, and Curvature Divergence.

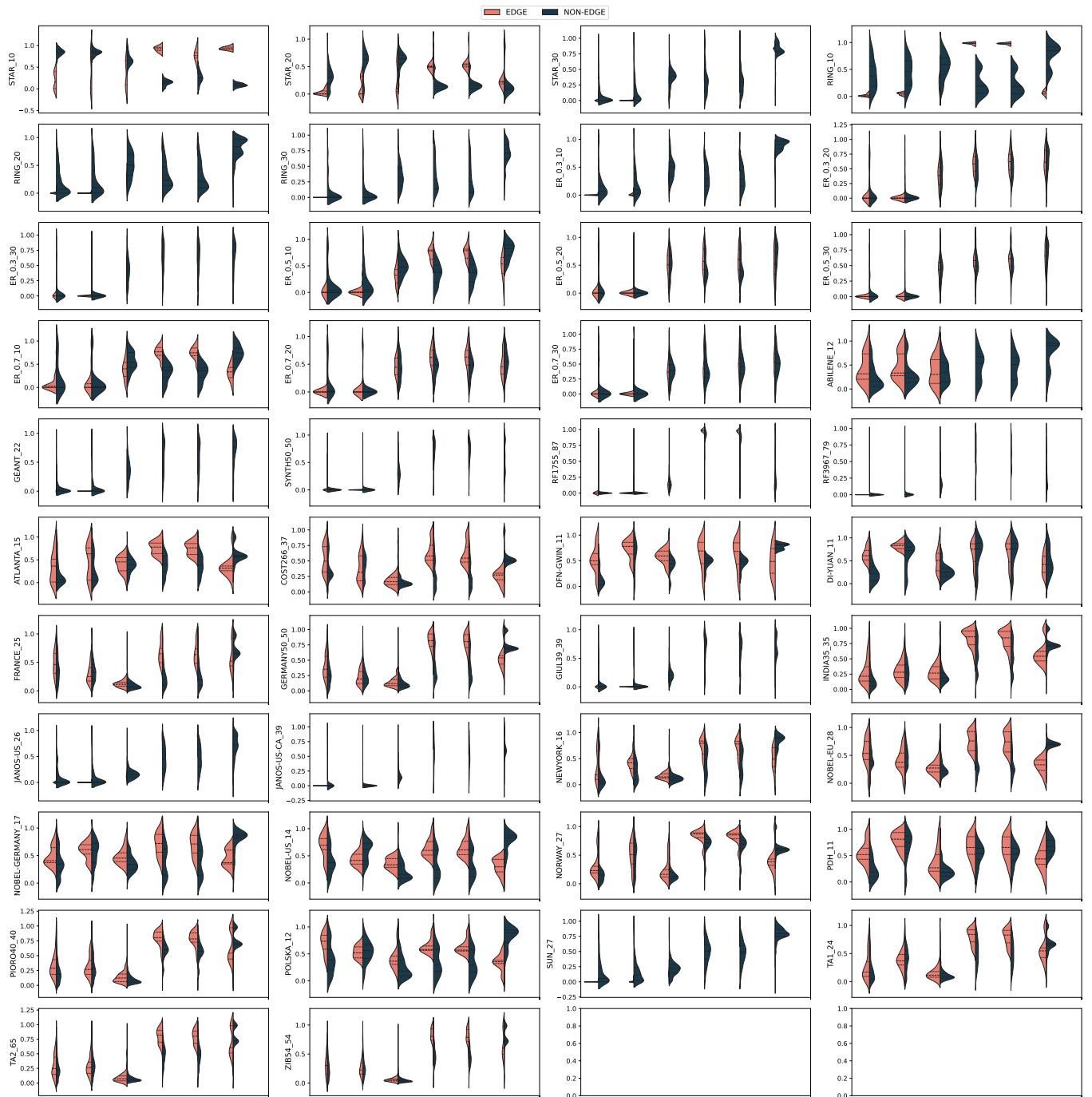


Figure 9. Normalized distributions of attack metrics in various topology DFL systems trained on the **ImageNet10** dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection.

In each subfigure, the attack metrics are respectively: Relative Loss, Relative Entropy, Relative Sensitivity, Cosine Similarity, Euclidean Similarity, and Curvature Divergence.

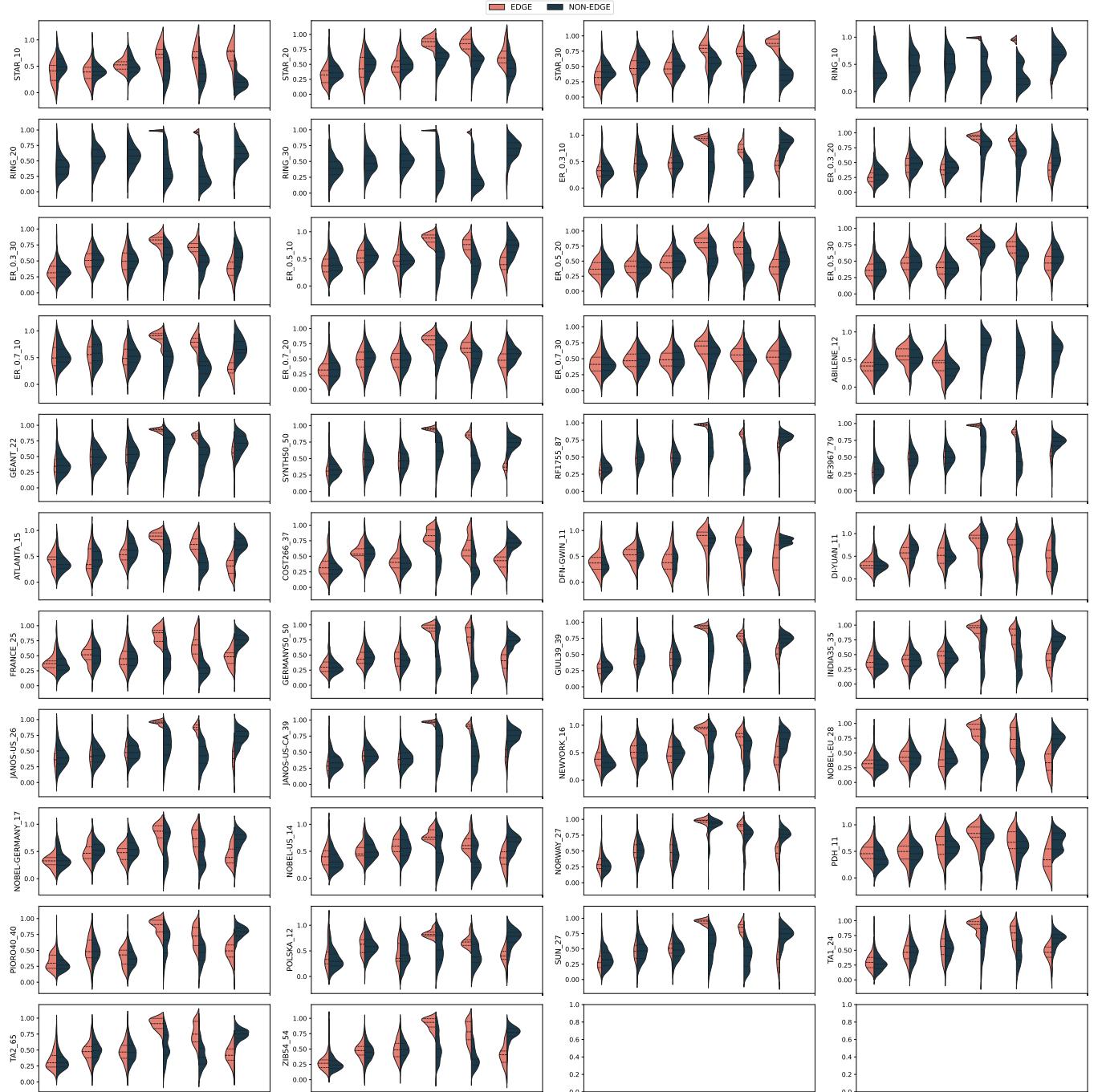


Figure 10. Normalized distributions of attack metrics in various topology DFL systems trained on the **Malware** dataset. The “edge group” corresponds to metrics calculated between directly connected nodes, while the “non-edge group” corresponds to metrics calculated between nodes with no direct connection. In each subfigure, the attack metrics are respectively: Relative Loss, Relative Entropy, Relative Sensitivity, Cosine Similarity, Euclidean Similarity, and Curvature Divergence.