Disrupting Cross-Community Information Flow in Decentralized Federated Learning

Xu Wang, *Student Member, IEEE*, Yuanzhu Chen, *Senior Member, IEEE*,
Qiang (John) Ye, *Senior Member, IEEE*, Jooyoung Son, and Octavia A. Dobre, *Fellow, IEEE*

***Abstract*—Structural disruptions in decentralized federated learning (DFL) can affect learning performance in different ways. In some cases, removing only a few nodes or edges causes severe performance degradation, whereas in others even extensive removals have limited effect. Severe performance degradation occurs when structural disruptions sever critical information flows, whereas disruptions that preserve these flows have limited impact on learning. Motivated by this observation, we propose a data-oriented community attack for DFL that explicitly targets critical information flows. The attack jointly leverages leaked client data patterns and network topology to identify data-oriented communities that shape information propagation in the network. Based on these communities, we develop a budget-constrained strategy for selecting node and edge removals that selectively disrupt information flow across communities. Experimental results show that the proposed attack strategy causes more severe performance degradation than structure-based baselines.**

***Index Terms*—Decentralized federated learning, adversarial attack, data-oriented communities, graph representation learning**

I. Introduction

Decentralized federated learning (DFL) relies on peer-to-peer communication to propagate model updates across a network [1]. In practice, structural removals do not necessarily degrade convergence in DFL. Removing a small number of nodes or edges can sometimes severely degrade convergence. In other cases, even extensive removals, including those at structurally central positions, have limited impact.

The impact of structural removals in decentralized federated learning depends on whether essential information flow is disrupted. This is governed by the interaction between network topology and non-independent and identically distributed (non-IID) data distributions across nodes. Real-world communication networks typically exhibit dense connectivity within communities, with only a limited number of links bridging different regions. In decentralized learning deployments, nodes within the same community therefore tend to hold relatively homogeneous data. As a result, many structurally important nodes or edges inside a community may carry largely redundant information, so their removal has limited impact on global learning dynamics. In contrast, sparse cross-community connections facilitate the propagation of complementary knowledge across heterogeneous regions. Removing even a small number of such links can therefore disproportionately impair global consensus and convergence.

Most existing approaches evaluate the impact of structural removals based solely on graph structure. Examples include degree [2], betweenness [3], closeness [4], collective influence [5], and message passing methods [6]. More recent extensions, such as spatial [7], geometric [8], and higher-order dismantling approaches [9], remain fundamentally grounded in structural information. Learning based methods, including reinforcement learning [10] and approaches that employ graph neural networks [11], have also been proposed. They learn removal strategies or node representations primarily from graph topology. When such methods are applied to DFL, they implicitly assume that structural importance aligns with learning importance.

In this work, we propose a data-oriented community attack for DFL that explicitly targets information flow across heterogeneous data regions under non-IID data distributions. We first construct data-aware node representations that integrate network topology with leaked data characteristics, enabling the identification of cross-community connectivity. We then develop a budget-constrained strategy for node and edge removals that disrupt information flow between heterogeneous regions. Experiments demonstrate that the proposed attack more effectively degrades learning performance than baselines under non-IID data distributions.

II. Data-Oriented Community attack Framework

A. Attack Setting and Design Rationale

We consider a decentralized federated learning system deployed over a fixed communication network, modeled as an undirected graph $G = (V, E)$, where nodes represent clients and edges represent bidirectional communication links. Each node exchanges model updates only with its neighbors.

We assume a structural adversary that observes the full network topology and infers coarse data characteristics from a small subset of compromised clients. Such partial data

The work of Xu Wang and Yuanzhu Chen is financially sponsored by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant, RGPIN-2017-05201. The work of Octavia A.Dobre was supported by NSERC Discovery Program under Grant RGPIN-2019-04123.

Xu Wang and Yuanzhu Chen are with the School of Computing, Queen's University, Kingston, ON K7L3N6 Canada. (e-mail: xu.wang@queensu.ca, yuanzhu.chen@queensu.ca)

Qiang (John) Ye is with the Department of Electrical & Software Engineering, University of Calgary, Calgary, AB, T2N1N4 Canada. (e-mail: qiang.ye@ucalgary.ca)

Jooyoung Son is with the Division of Marine System Engineering, Korea Maritime & Ocean University, Busan, 49112 Korea. (e-mail: mmlab@kmou.ac.kr)

Octavia A. Dobre is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL A1C5S7 Canada. (e-mail: odobre@mun.ca)

knowledge is reasonable, as prior work has shown that local training data can be inferred through gradient leakage and membership inference attacks [12]–[14], and large-scale device compromises have been widely reported in practice, as exemplified by the Mirai botnet [15].

Operating under a budget B, the adversary can disable selected nodes or edges to impair learning. Disabling a node u incurs a cost $c_1(u)$ and implicitly removes all its incident edges, whereas disabling an edge e incurs a cost $c_2(e)$. The adversary is limited to connectivity disruptions only, cannot alter models or data, and performs a one-time intervention prior to training. This setting reflects reasonable scenarios involving device failures, maintenance actions, or targeted link disruptions, with learning algorithms and local datasets remaining intact.

The key motivation is that decentralized learning efficiency depends on non-local information exchange. Under heterogeneous data, clients with similar data properties form implicit coordination regions that do not necessarily align with the network topology. Links bridging these regions are critical for propagating complementary information. Topology-only dismantling methods overlook this structure. They often target structurally central components with limited impact on learning, while leaving intact the pathways that support cross-region coordination. These insights motivate a data-oriented attack strategy that jointly accounts for network structure and data heterogeneity under a budget constraint.

B. Revealing Data-Oriented Communication Structure

To disrupt communication pathways that govern information propagation in decentralized learning, the attacker must first uncover how information is organized across the network. We assume the attacker has limited access to client data patterns through a small set of compromised nodes. In decentralized learning systems, client compromise can occur due to insecure edge devices, adversarial participants, or unintended information exposure. Based on these partial observations, we construct a data-aware representation of the network. Each node is assigned a feature vector that combines structural information with data-pattern information observed at compromised nodes. Nodes without direct data observations acquire data-related signals indirectly through neighborhood aggregation, allowing data influence to propagate beyond compromised locations.

We adopt a message-passing encoder to aggregate these signals across the graph. Through multi-hop propagation, nodes that occupy similar communication roles or exhibit similar inferred data behavior become close in the learned representation space. This process produces embeddings that reflect both how nodes are connected and how information associated with different data patterns flows through the network.

Data-oriented communities are then identified by clustering the learned embeddings. Let $V = V_1 \cup \cdots \cup V_p$ denote the resulting partition of nodes into data-oriented communities, where each V_i groups nodes with similar structural roles and inferred data patterns. Unlike structural communities, these data-oriented communities capture information propagation units that reveal vulnerable cross-community connections.

C. Community-Level Vulnerability Characterization

Once data-oriented communities are identified, the next step is to characterize their vulnerability under structural disruption. In decentralized federated learning, nodes within the same community share similar data characteristics and rely on cross-community connections to acquire complementary information. These cross-community links enable non-local information propagation across heterogeneous data regions. Disrupting them directly limits the ability of nodes to learn from data held outside their own community, making them critical targets for attack.

To quantify the strength of coupling between a community and the rest of the network, we adopt conductance as a community-level metric. For a given community V_i, let $\mathrm{cut}(V_i, \bar{V}_i)$ denote the total number of edges between V_i and its complement $\bar{V}_i$, and let $\mathrm{vol}(V_i)$ denote the sum of degrees of nodes in V_i. The conductance of V_i is defined as [16]

$$\phi(V_i) = \frac{\mathrm{cut}(V_i, \bar{V}_i)}{\min\{\mathrm{vol}(V_i),\, \mathrm{vol}(\bar{V}_i)\}}. \tag{1}$$

Low conductance indicates that a community is well separated from the rest of the network, while high conductance reflects a stronger reliance on cross-community communication. In DFL, conductance therefore serves as a proxy for cross-community information flow. Reducing conductance weakens the channels through which complementary data are exchanged, increasing the vulnerability of the learning process to structural disruption.

This interpretation naturally guides the subsequent attack strategy. Rather than dismantling the network indiscriminately, the attacker seeks to selectively reduce conductance between data-oriented communities under a budget constraint.

D. Budget-Constrained Data-Oriented Community Disruption

Having identified data-oriented communities and characterized their vulnerability through conductance, we now describe how the attacker selects node and edge removals under a fixed budget. The objective is to weaken communication across data-oriented communities while avoiding unnecessary disruption within communities.

We introduce binary decision variables to represent removal actions. For each node $u \in V$, let $x_u = 1$ indicate that node u is removed. For each edge $e \in E$, let $y_e = 1$ indicate that edge e is removed explicitly. An edge $e = (u, v)$ becomes inactive if either endpoint is removed or if the edge itself is removed. We capture this effect using

$$l_{(u,v)} = \max\{x_u,\, x_v,\, y_{(u,v)}\}, \tag{2}$$

where $l_{(u,v)} = 1$ indicates that edge (u, v) is no longer available for communication.

To guide the attack toward community-level disruption, we construct an objective that targets cross-community communication while preserving internal structure. This objective is

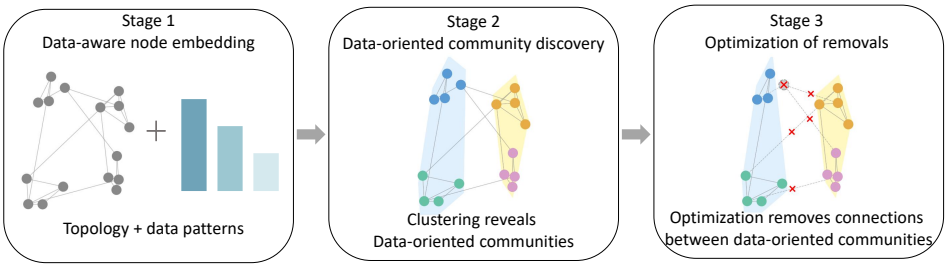


Fig. 1: Overall process of the proposed data-aware attack framework.

built from two components that approximate inter-community conductance.

For each data-oriented community V_i, the first component penalizes the removal of high-degree nodes within the community,

$$m_1(V_i) = \frac{\sum_{u \in V_i} d_u \, x_u}{\sum_{u \in V_i} d_u}, \tag{3}$$

which discourages unnecessary reduction of internal connectivity. Here, d_u denotes the degree of node u, and $c_1(u)$ and $c_2(e)$ represent the node and edge removal costs, respectively, as defined in the previous section. The second component measures the loss of edges connecting V_i to the rest of the network,

$$m_2(V_i) = \sum_{\substack{(u,v)\in E:\\ u\in V_i,\, v\notin V_i}} l_{(u,v)}, \tag{4}$$

which corresponds to the cut term in conductance.

Together, $m_1(V_i)$ and $m_2(V_i)$ form a surrogate for community conductance. Minimizing $m_2(V_i)$ reduces cross-community connectivity, while controlling $m_1(V_i)$ prevents trivial solutions that rely on removing large internal hubs.

In addition to the attack objective, the attacker is constrained by a limited removal budget. We capture budget usage using the total removal cost

$$m_3 = \sum_{u \in V} c_1(u)\, x_u + \sum_{e \in E} c_2(e)\, y_e. \tag{5}$$

Removing a node incurs only the node-removal cost, while any incident edges disabled as a consequence incur no additional cost. In contrast, explicitly removing an edge incurs the corresponding edge-removal cost.

Aggregating these quantities across all communities yields the following budget-constrained optimization problem:

$$\text{(P1)} \quad \min_{x,\,y} \left(\sum_{i=1}^{p} m_1(V_i) + \lambda \sum_{i=1}^{p} m_2(V_i) + \varepsilon\, m_3 \right) \tag{6a}$$

$$\text{s.t.} \quad m_3 \le B. \tag{6b}$$

The parameter λ balances the emphasis between preserving internal community structure and disrupting inter-community communication. The regularization term weighted by ε discourages unnecessary budget expenditure when further removals provide limited additional benefit.

We solve problem (P1) using the COIN-OR CBC solver (Computational Infrastructure for Operations Research), an open-source branch-and-cut mixed-integer programming solver [17]. While this exact optimization approach yields optimal solutions, its computational complexity grows exponentially with network size due to NP-hardness. A potential scalable alternative is to adopt greedy heuristics that incrementally select nodes or edges to improve network conductance under cost constraints.

E. Overall Attack Procedure

The proposed attack follows a three-stage pipeline, as illustrated in Fig. 1. First, the attacker constructs a data-aware representation of the network by integrating the communication topology with partial data patterns obtained from compromised nodes. This representation captures how information associated with heterogeneous data propagates through the network.

Second, data-oriented communities are identified by clustering the learned node representations. These communities group nodes that play similar communication roles and exhibit similar inferred data characteristics, even when they are not densely connected in the graph. The resulting partition reveals structures that govern non-local information exchange.

Finally, the attacker performs a budget-constrained selection of node and edge removals. By targeting communication links that connect different data-oriented communities, the attack selectively disrupts the pathways that support information propagation across heterogeneous data regions.

III. Experimental Evaluation

We evaluate the proposed method on a synthetic four-community network with non-IID MNIST data [18]. The experimental setup specifies the network structure, data partitioning, attack cost model, and the construction of data-aware embeddings and communities. Table I summarizes the experimental configuration.

We first examine the behavior of the proposed method under different budget levels. We then show how data-aware embeddings identify data-oriented communities that guide boundary targeting. Next, we compare the proposed method with topology-only dismantling baselines by measuring conductance reduction and analyzing node and edge removal behavior. Finally, we evaluate the effect of the proposed method on DFL learning performance.

TABLE I: Summary of Experimental Parameters

Category	Parameters
1. Synthetic Network Configuration	
Number of nodes	100 (4 communities, 25 nodes each)
Intra-community topology	Community 0: Barabási–Albert (BA) ($m = 4$); Community 1: Complete graph; Community 2: Watts–Strogatz ($k = 6$, $p = 0.3$); Community 3: BA ($m = 4$)
Inter-community edges	All-to-all community pairs, 3 random edges per pair
2. Data Distribution	
Dataset	MNIST (60k train, 10k test)
Non-IID partition	Communities 0 and 2: digits 0–4; Communities 1 and 3: digits 5–9
Samples per node	600 training samples; centralized test set
Compromised nodes	12 nodes (3 per community) used for data leakage
3. Attack Cost Model	
Node costs	Boundary: $\mathcal{N}(8.0, 2.0^2)$; Internal: $\mathcal{N}(5.0, 1.5^2)$
Edge costs	Inter-community: $\mathcal{N}(6.0, 1.5^2)$; Intra-community: $\mathcal{N}(3.0, 1.0^2)$
4. GraphSAGE Embedding	
Node features	Degree ($\log(1 + d)$), clustering coefficient, PageRank ($\log(1 + r)$), k-core
Positional encoding	Laplacian eigenvectors 2–9 (8 dims), sign aligned
Model architecture	2-layer GraphSAGE, 96-d hidden layers, mean aggregation
Training	AdamW (learning rate $l_r = 10^{-3}$, weight decay $w_d = 10^{-4}$), 300 epochs
Dropout/Activation	Dropout 0.5, ReLU, batch normalization
Output embeddings	96-dimensional, L2-normalized
5. Clustering Configuration	
Method	Agglomerative hierarchical clustering
Distance/Linkage	Euclidean distance, Ward linkage
Cluster range	$k \in [2, 15]$
Objective	Penalized silhouette: $\text{Silhouette}(k) - 0.02k$
Post-processing	Clusters with fewer than 10 nodes merged to nearest larger cluster

A. Experimental Settings

We evaluate the proposed method in a controlled decentralized learning environment consisting of a synthetic network, a highly non-IID data distribution, and a heterogeneous attack cost model. All experimental parameters are summarized in Table I.

We construct a 100-node synthetic network partitioned into four communities with distinct intra-community topologies and sparse cross-community connectivity, as illustrated in Fig. 2. MNIST is partitioned into a strongly non-IID configuration in which different communities hold disjoint digit subsets. A small set of nodes is compromised to provide partial data leakage for embedding construction. Node and edge removal costs are drawn from heterogeneous distributions, with higher expected costs assigned to boundary nodes and inter-community edges.

Each node is encoded using a GraphSAGE-based data-aware embedding that integrates structural information with

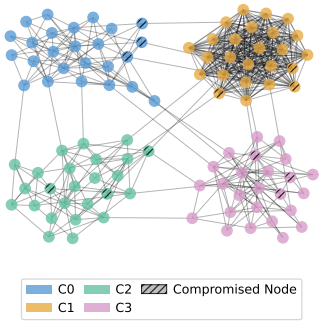


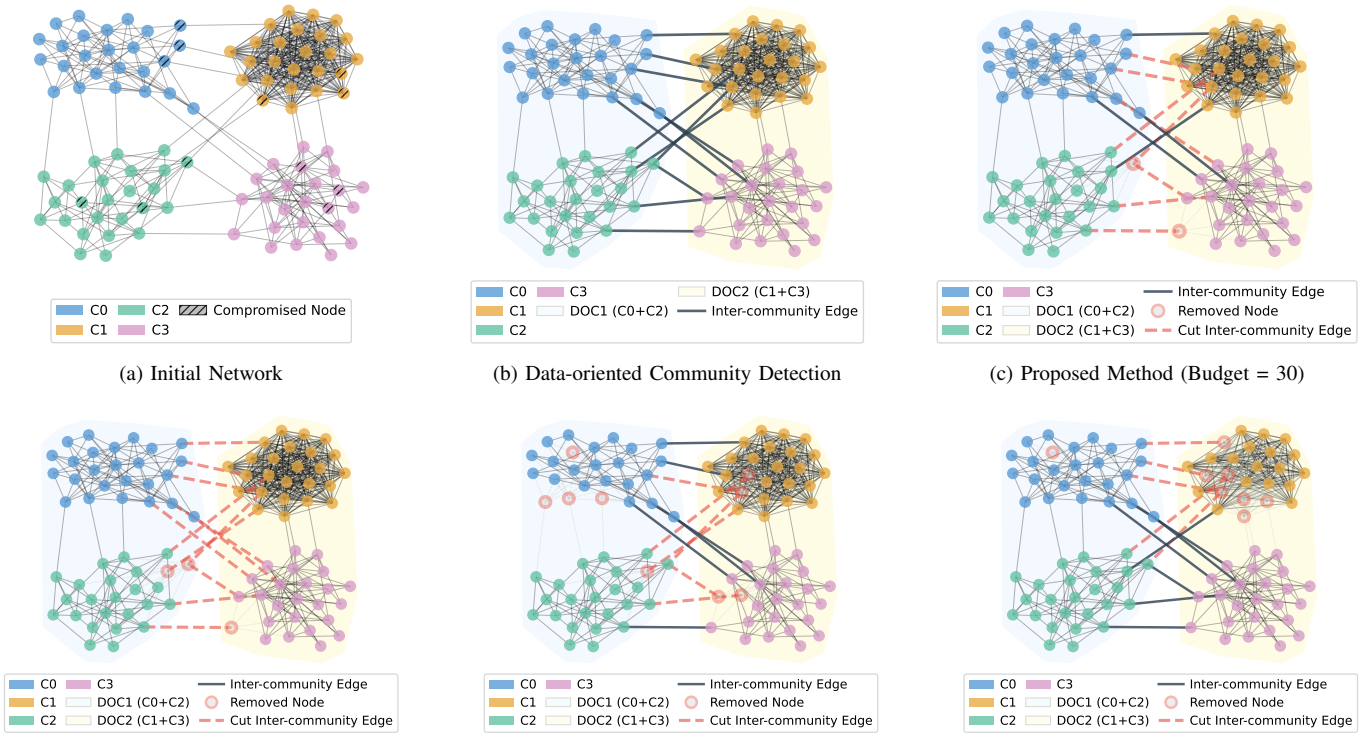
Fig. 2: Synthetic network topology illustrating the four communities and cross-community connectivity.

Laplacian positional encodings. Agglomerative hierarchical clustering is applied to the learned embeddings to identify data-oriented communities that guide boundary targeting.

B. Attack Behavior Under Different Budget Levels: An Example

We use a simple example to show how nodes and edges are removed during the attack. Fig. 3 summarizes the proposed method and compares it with baseline methods. The example starts from the initial information used in the attack, shown in Fig. 3a, which includes the full 100-node network and data obtained from 12 compromised nodes.

Based on this information, data-aware community detection is applied, as shown in Fig. 3b. Node embeddings combine network topology with data profiles obtained from compromised nodes. Clustering of these embeddings separates the network into two data-oriented groups: DOC1, formed by communities C0 and C2, and DOC2, formed by communities C1 and C3 [19]. Edges linking DOC1 and DOC2 are identified as cross-community connections that enable information exchange between the two communities. Since DOC1 contains only digits 0–4 and DOC2 contains only digits 5–9, these connections provide the sole paths for sharing complementary data across the network. Fig. 3c and 3d show the nodes and edges removed by the proposed method at budgets $B = 30$ and $B = 60$. At $B = 30$, two nodes and eight edges are removed between DOC1 and DOC2, leaving only a small number of cross-community links. At $B = 60$, three nodes and twelve edges are removed between the data-oriented communities, which eliminates all remaining connections and results in a complete separation. Fig. 3e and 3f report the baseline attack results at budget $B = 60$. The betweenness-based baseline removes nine nodes but only weakens cross-community connectivity. Since the selection relies purely on topology, many removals occur on internal links within each data-oriented community, for example between C0 and C2. Consequently, a large fraction of cross-community edges remains, and the two communities are not fully separated. The high-degree baseline removes eight nodes and shows a similar weakness. Because Community 1 is a complete graph, its nodes have consistently high degrees and are frequently selected. These removals



(a) Initial Network

(b) Data-oriented Community Detection

(c) Proposed Method (Budget = 30)

(d) Proposed Method (Budget = 60)

(e) Betweenness Baseline (Budget = 60)

(f) High-Degree Baseline (Budget = 60)

Fig. 3: Comparison of attack methods showing node removal strategies and their impact on inter-data-oriented community connectivity.

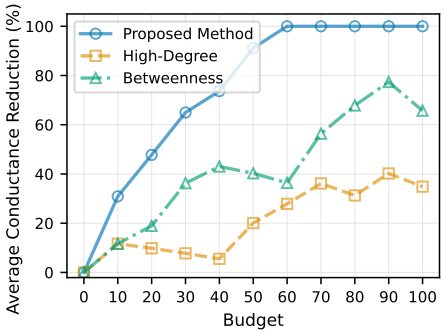


Fig. 4: Conductance reduction over different budget.

mainly affect internal links and do not reduce DOC1–DOC2 connectivity. As a result, much of the budget is spent with limited impact, and the two data-oriented communities remain partially connected.

C. Conductance Comparison and Removal Behavior

Fig. 4 compares conductance reduction under different budgets. The proposed method reduces conductance more effectively than both baselines across the entire budget range and reaches full separation at a budget of 60. In contrast, the High-Degree and Betweenness baselines show non-monotonic changes as the budget increases. At budget 100, conductance reduction reaches 34.83% for High-Degree and 65.72% for Betweenness, and neither baseline fully disconnects the communities. Baseline instability arises from greedy selection. High-Degree frequently removes nodes in dense communities, yielding limited reductions unless boundary nodes are chosen. Betweenness fluctuates because each removal reshapes shortest paths, causing alternating selections between cross-community and internal nodes. In contrast, the proposed method first identifies data-oriented communities using data-aware embeddings. Node and edge removals are then selected jointly to focus on the links between these communities under the budget constraint.

Fig. 5 summarizes node and edge removals as the budget increases. Node removals are shown on the left y-axis, while edge removals are shown on the right y-axis, with stacked bars separating explicit and implicit removals. The removal process follows three stages. From budget 0 to 20, the method gradually removes two nodes and a growing number of edges, reaching 16 removed edges by budget 20, most of which are implicit due to node deletions. Between budgets 20 and 50, the number of removed nodes remains fixed, while additional budget is used to remove edges explicitly, indicating a shift toward edge-focused removals when further node deletions become inefficient. A transition occurs between budgets 50 and 60, where a third node is removed, causing a sharp

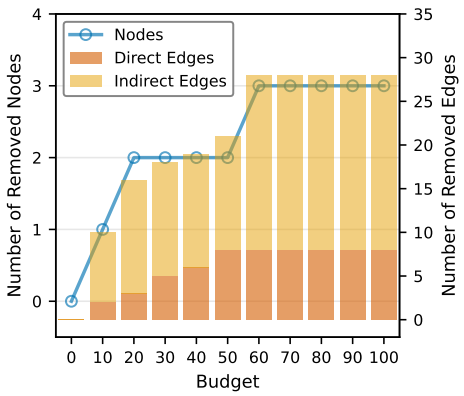


Fig. 5: Number of removed nodes and edges over different budget.

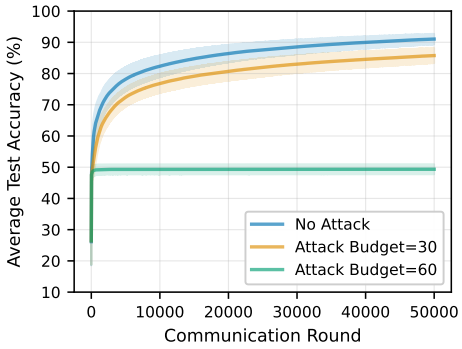


Fig. 6: Average test accuracy over different communication round.

increase in implicitly removed edges. Beyond budget 60, all removal counts remain unchanged, indicating that full separation is achieved and maintained for larger budgets.

D. Impact of Network Disruptions on Federated Learning

Fig. 6 shows the evolution of average test accuracy over 50,000 communication rounds under three settings: no attack, budget 30, and budget 60. Without an attack, training converges steadily and reaches an accuracy above 90%. With a budget of 30, convergence remains stable but the final accuracy is slightly lower, as residual connections still allow limited information exchange between the two data-oriented communities. In contrast, a budget of 60 fully separates the network into two isolated parts. Once cross-community communication is removed, each partition trains only on its local data, and the accuracy quickly saturates with no further improvement.

IV. Conclusion and Future Work

In DFL, information propagation is jointly shaped by network connectivity and heterogeneous client data, creating exploitable vulnerabilities in how information flows across the network. These vulnerabilities manifest as data-oriented communities that structure cross-community information flow. By integrating leaked data patterns with network topology, we identify these communities and perform budget-constrained node and edge removals to selectively sever information flow between them. As a result, decentralized learning performance degrades substantially.

The proposed attack considers a static setting and assumes fixed costs and availability of nodes and edges, without modeling temporal variations or adaptive responses. An important direction for future work is to extend this framework to dynamic and adaptive settings, where costs, network availability, and attacker–defender interactions evolve over time.

References

[1] X. Wang, Y. Chen, Q. Ye, and O. A. Dobre, "Teleportation links: Mitigating catastrophic forgetting in decentralized federated learning," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 2167–2180, Sept. 2026.

[2] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.

[3] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Physical Review E*, vol. 65, p. 056109, May 2002.

[4] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang, "Attack robustness and centrality of complex networks," *PLOS ONE*, vol. 8, no. 4, pp. 1–17, Apr. 2013.

[5] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, Oct. 2015.

[6] A. Braunstein, L. Dall'Asta, G. Semerjian, and L. Zdeborová, "Network dismantling," *Proceedings of the National Academy of Sciences*, vol. 113, no. 44, pp. 12 368–12 373, Oct. 2016.

[7] Z.-G. Wang, Y. Deng, Z. Wang, and J. Wu, "Disintegrating spatial networks based on region centrality," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 6, p. 061101, Jun. 2021.

[8] S. Osat, F. Papadopoulos, A. S. Teixeira, and F. Radicchi, "Embedding-aided network dismantling," *Physical Review Research*, vol. 5, p. 013076, Feb. 2023.

[9] P. Peng, T. Fan, and L. Lü, "Network higher-order structure dismantling," *Entropy*, vol. 26, no. 3, p. 248, Mar. 2024.

[10] C. Fan, L. Zeng, Y. Sun, and Y.-Y. Liu, "Finding key players in complex networks through deep reinforcement learning," *Nature machine intelligence*, vol. 2, no. 6, pp. 317–324, May 2020.

[11] M. Grassia, M. De Domenico, and G. Mangioni, "Machine learning dismantling and early-warning signals of disintegration in complex systems," *Nature communications*, vol. 12, no. 1, p. 5190, Aug. 2021.

[12] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems (NeurIPS)*, vol. 32, Dec. 2019.

[13] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 691–706.

[14] 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)*, San Jose, CA, USA, Jun. 2017, pp. 3–18.

[15] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*, Vancouver, BC, Aug. 2017, pp. 1093–1110.

[16] L. Lin, R.-H. Li, and T. Jia, "Scalable and effective conductance-based graph clustering," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*, Washington, DC, USA, Feb. 2023.

[17] COIN-OR Foundation, "COIN-OR: Computational Infrastructure for Operations Research," https://www.coin-or.org/.

[18] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Oct. 2012.

[19] A.-L. Barabási and M. Pósfai, *Network science*. Cambridge, England: Cambridge University Press, Jul. 2016.