User Isolation Poisoning on Decentralized Federated Learning: An Adversarial Message Passing Graph Neural Network Approach

Kai Li, *Senior Member, IEEE*, Yilei Liang, *Student Member, IEEE*, Pietro Liò, *Fellow, IEEE*, Wei Ni, *Fellow, IEEE*, Falko Dressler, *Fellow, IEEE*, Jon Crowcroft, *Fellow, IEEE*, and Ozgur B. Akan, *Fellow, IEEE*

***Abstract*—This paper proposes a new cyberattack on Decentralized Federated Learning (DFL), named User Isolation Poisoning (UIP). While following the standard DFL protocol of receiving and aggregating benign local models, a malicious user strategically generates and distributes compromised updates to undermine the learning process. The objective of the new UIP attack is to diminish the impact of benign users by isolating their model updates, thereby manipulating the shared model to reduce the learning accuracy. To realize this attack, we design a novel threat model that leverages an adversarial Message Passing Graph (MPG) neural network. Through iterative message passing, the adversarial MPG progressively refines the representations (also known as embeddings or hidden states) of each benign local model update. By orchestrating feature exchanges among connected nodes in a targeted manner, the malicious users effectively curtail the genuine data features of benign local models, thereby diminishing their overall influence within the DFL process. The MPG-based UIP attack is implemented in PyTorch, demonstrating that it effectively reduces the test accuracy of DFL by 49.5%, and successfully evades existing cosine similarity-based and Euclidean distance-based defense strategies.**

***Index Terms*—User Isolation, Poisoning Attack, Decentralized Federated Learning, Model Correlations, Message Passing Graph Neural Networks**

I. Introduction

Decentralized federated learning (DFL), as a distributed machine learning, enables a network of user devices to collaboratively train a shared model without a central coordinating server that aggregates model updates from individual users [1]. The decentralized variant eliminates this central point by allowing users to communicate directly with each other in a peer-to-peer fashion. Each user device maintains its local dataset, ensuring that sensitive information remains on-premise and enhancing privacy compliance [2]. The collaborative process involves exchanging model parameters or gradients with neighboring users, often structured in a network topology, such as a graph, to iteratively improve the shared model collectively.

DFL has a significant potential to enhance connected services by enabling secure, collaborative machine learning without compromising user privacy. For example, in the immersive virtual environments of the Metaverse, users generate extensive health-related data through interactions with wearable devices [3], biometric sensors [4], and virtual health applications [5]. DFL allows this sensitive data to remain with user devices or local nodes while contributing to the training of global health models [6]. By aggregating only model updates rather than raw data, DFL ensures that personal health information is not exposed or transmitted across the network.

Due to the absence of a central server and the reliance on peer-to-peer update exchanges, model poisoning in DFL poses a broad spectrum of security threats, where adversaries can exploit local trust relationships to manipulate the training process. Typical threats include Byzantine poisoning, where malicious users inject arbitrary gradients to destabilize consensus [7]; collusion-based attacks, where groups of adversaries coordinate to amplify their influence [8]; free-rider behaviors, where participants contribute no useful updates but still benefit from the global model [9]; and isolation attacks, where adversaries suppress or marginalize the impact of benign users' contributions. These threats can degrade DFL accuracy, bias model behavior, or undermine fairness without breaching data privacy.

In this paper, we propose a novel cyberattack on DFL, named *user isolation poisoning (UIP)*, which exploits the collaborative yet decentralized nature of DFL while following the standard protocol of receiving and aggregating benign local model updates from neighboring users. Unlike benign users, a malicious user generates compromised model updates intentionally designed to be shared with peers to disrupt the learning process of DFL.

The primary objective of the UIP attack is to isolate the model updates of benign users, thereby impairing their ability to contribute to the shared model. To implement this attack, we propose a new threat model based on an

K. Li is with the Department of Information Technology, Kennesaw State University, Marietta, GA 30060, and also with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA (E-mail: kaili@ieee.org).

Y. Liang, P. Liò, and J. Crowcroft are with Department of Computer Science and Technology, University of Cambridge, CB3 0FA Cambridge, UK. J. Crowcroft is also with The Alan Turing Institute, London, UK. (Email: {yl841, pl219}@cam.ac.uk, jon.crowcroft@cl.cam.ac.uk).

W. Ni is with the Digital Productivity and Services Flagship, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, NSW 2122, Australia (E-mail: wei.ni@data61.csiro.au).

F. Dressler is with the Telecommunication Networks group (TKN) at the School of Electrical Engineering and Computer Science, TU Berlin, Germany (E-mail: dressler@ccs-labs.org).

O. B. Akan is with the Division of Electrical Engineering, Department of Engineering, University of Cambridge, CB3 0FA Cambridge, U.K., and also with the Center for NeXt-Generation Communications (CXC), Koç University, 34450 Istanbul, Turkey (E-mail: oba21@cam.ac.uk).

adversarial message passing graph neural network (MPG), which enables a malicious user to generate compromised model updates. The proposed adversarial MPG iteratively refreshes the representation (a.k.a. embeddings or hidden states) of each benign model update by exchanging feature information with its neighbors through a message-passing mechanism [10]. Instead of aggregating all neighbors' model updates, it introduces biased propagation weights that subtly distort the correlations between malicious and benign updates. This selective tailoring enables the attacker to diminish the influence of benign users while amplifying its adversarial contributions, thereby isolating benign contributions from the evolving shared model. The distortions are carefully crafted so that the malicious model updates remain statistically consistent with benign model updates and can evade detection by similarity- or distance-based defenses. The malicious user can craft the compromised model updates to minimize the test accuracy of DFL; the shared model can be manipulated towards the malicious user's objectives while diminishing the influence of legitimate data contributions.

The benign users can employ model poisoning detection techniques on the local model updates from their peers, scrutinizing them for statistically significant deviations or anomalies that may indicate malicious alterations. Measurement metrics, such as cosine similarity or Euclidean distance, can be computed to identify model updates that significantly diverge in direction or magnitude. To bypass the detection of existing defense models, the proposed MPG-based UIP attack tailors the malicious user's adversarial model updates, thus maintaining compatibility with their benign counterparts while undermining the DFL.

The key contributions of this paper are as follows:

- The new UIP attack is proposed to intentionally isolate the benign users' model updates, which can result in a manipulated shared model and diminish the influence of benign data contributions. A new architecture is designed to synthesize compromised model updates to effectively minimize the test accuracy of DFL, thus skewing the learning process away from accurate representations of the benign users' data.
- As the optimization of the adversarial training model at a malicious user is a challenging non-convex combinatorial problem, a new graph signal processing approach is developed to iteratively optimize the compromised model updates by running the UIP and sub-gradient descent alternately.
- Within the UIP architecture, the adversarial MPG is trained alongside sub-gradient descent to capture the interactions among the benign users' model updates. By manipulatively reconstructing the correlations of these model updates, the adversarial MPG aims to maximize the reconstruction loss while keeping the compromised model updates undetectable.

The proposed MPG-based UIP attack is implemented in PyTorch, showing experimentally that the MPG-based UIP attack successfully reduces the test accuracy by 49.5%, and bypasses the detection of existing cosine similarity- and Euclidean distance-based defense models. The source code of the MPG-based UIP attack is released on GitHub: https://github.com/AnonymousAuthors/DFL_Attack.

The proposed UIP attack demonstrates that an adversary can compromise the integrity of collaborative training by isolating some benign users' contributions and skewing the shared model, even without access to raw data. While DFL preserves data locality and prevents raw data leakage, no privacy in DFL can be guaranteed under the UIP attack since adversaries can exploit correlations in exchanged updates to manipulate outcomes. Moreover, the limitations of existing defenses are exposed in peer-to-peer settings: By strategically weakening the influence of benign updates, the attack renders training ineffective and undermines accuracy across participants. The UIP attack is effective within DFL contexts, as it simultaneously preserves an appearance of normalcy to benign peers and evades detection.

In addition, the proposed MPG-based UIP attack targets DFL scenarios in which benign users train and share the same underlying model architecture, implying homogeneous data features across the network. By isolating benign users' model updates, the malicious user exploits feature-level correlations to reduce the diversity and richness of the data features that the shared model learns from, leading to a degraded DFL accuracy. The exploration of MPG-based UIP attacks underpins the need for rigorous investigation into safeguard mechanisms to defend against such subtle and impactful adversarial undertakings in DFL.

The remainder of this paper is structured as follows. Section II reviews the literature on poisoning threats against DFL. Section III investigates the DFL training process with benign users as well as state-of-the-art cosine similarity-based and Euclidean distance-based defense models. The proposed MPG-based UIP attack is described in Section IV. Section V presents the performance analysis. Section VI concludes the paper.

II. Literature Review on Poisoning Attacks

This section reviews the literature on poisoning threats to DFL and centralized FL (CFL) in a comparison with the new MPG-based UIP attack developed in this paper.

For instance, in [8], a collusion-based poisoning attack on DFL was studied, where malicious local models are assessed by computing the Euclidean distance from benign models and assigning a toxicity score. The model exhibiting the largest deviation (i.e., the highest toxicity score) is selected by the attacker as the poisoning model update. However, this approach demands tight coordination among all attackers so that they can converge on the same malicious model. A falsified-data-based poisoning attack that injects deceptive data points into the training process of the blockchain-enabled DFL was studied in [11]. By exploiting blockchain's distributed ledger for storing and exchanging model parameters, the attacker can generate falsified data to permeate the learning process.

The study in [7] employed a conventional adversarial framework on DFL in which coordinated attackers corrupt

local datasets or craft adversarial local model updates for propagation to neighboring users. This threat model allows the attacker to selectively distribute manipulated model parameters to connected peers, with the flexibility to use distinct malicious local models for individual neighbors. In [12], a poisoning framework was presented for DFL that compromises the integrity of benign users' local models. To identify such attacks, users compute pairwise Euclidean distances between their own local models and those of neighboring peers, leveraging these metrics to derive trust scores for assessing model legitimacy.

To mitigate the model poisoning attack in CFL, several robust aggregation methods have been proposed, such as coordinate-wise median [13], geometric median, RFA [14], and FoolsGold [15]. These methods rely on the server having access to the full set of client updates to compute robust statistics or detect Sybil behaviors, which is challenging or even not possible in DFL due to the lack of a central server. While [16] is designed to mitigate the model poisoning attack in DFL, it still relies on the Euclidean distance and Cosine similarity to detect anomalies.

Based on several existing poisoning attacks in CFL, a range of parameters relevant to attacking efficiency can be defined [17]. Using these attacks, the defense models could be evaluated to understand the assumptions and defensive outcomes. The authors of [18] introduced a model poisoning strategy for CFL that leverages malicious users disguised as benign participants. These malicious users falsify local models to poison the shared model, enabling stealth backdoor injection or learning efficacy degradation of the CFL. In addition, CFL relying on weighted or trimmed averaging defense frameworks remains susceptible to stealthy poisoning attacks [19], which induce accuracy degradation during training. An attack model against CFL was designed by exploiting the inherent properties of CFL protocols and their aggregation mechanisms to inject malicious models into CFL.

Recent works [20], [21] investigated data-agnostic model poisoning in CFL. In these settings, benign users transmit their local models to a central server, and the adversary passively intercepts shared updates from neighboring clients. Their threat models employ graph autoencoder (GAE) architectures to regenerate the structural correlations of model weights and then craft malicious local models that maximize the global training loss. These approaches rely heavily on weight-level manipulation under a server-coordinated learning paradigm.

Unlike these existing studies, this paper explores UIP attacks in DFL, where no central server exists and the model updates are exchanged among peers. As summarized in Table I, the proposed MPG-based UIP attack introduces a holistic strategy that enables a more covert and effective disruption of DFL than existing poisoning methods focused on collusion-based manipulation or malicious local model crafting, e.g., [8], [20], [21]. These existing attacks typically rely on conspicuous deviations in model parameters or injected anomalies that can be identified by the defense mechanisms based on Euclidean distance or cosine similarity. In contrast, the UIP attack represents a fundamentally different threat by deliberately isolating the contributions of benign users, thereby suppressing their influence on the shared model. This targeted isolation skews the DFL process, reducing test accuracy without introducing detectable abnormalities. To achieve this, we adopt a novel framework that iteratively refines malicious updates through an alternating optimization procedure involving UIP execution and sub-gradient descent. The framework also integrates an adversarial MPG module that manipulates the structural correlations among benign model updates, effectively maximizing reconstruction loss while preserving stealth to evade conventional distance/similarity-based defenses.

TABLE I: Typical attacks against FL

	Characteristics
Collusion-based manipulation [8]	Attackers are coordinated to generate malicious local models by computing the Euclidean distance from benign models and assigning a toxicity score.
GAE-based poisoning [20], [21]	Benign users upload the local models to a central server to train a global model, while the attacker passively intercepts the benign model updates from its neighbors and synthesizes malicious local models. GAE is applied to regenerate the structural correlations of model weights and subsequently craft malicious local models.
Proposed MPG-based UIP attack	The new A-MPNN introduces iterative message-passing layers that explicitly model feature-level interactions across neighboring updates in the DFL graph. Dependencies that are invisible to the GAE can be learned and manipulated, thereby producing UIP model updates that more effectively isolate benign users.

TABLE II: Notation and definition

Notation	**Definition**
N	The total number of benign users
$\boldsymbol{\omega}_n(\mathcal{T})$	The local model parameters of user n
$\boldsymbol{\omega}_n^S(\mathcal{T})$	The shared model of DFL in the $\mathcal{T}$-th communication round
N'	The number of authorized (legitimate) but malicious users
$\boldsymbol{\omega}_j^a(\mathcal{T})$	The model update of the malicious user j
$\mathcal{E}(\boldsymbol{\omega}_n(\mathcal{T}); x(a_n), y(a_n))$	The training loss function of benign user n
$\overline{\omega}_{i',i''}$	The cosine similarity between the two user's model updates
$d(\delta(\boldsymbol{\omega}_{i'}^S(\mathcal{T}), \delta(\boldsymbol{\omega}_{i''}^S(\mathcal{T}))$	the Euclidean distance between any two shared model updates
d_T^{Cos}	Cosine similarity threshold
d_T^{Euc}	Euclidean distance threshold
η	The adjacency matrix for the local models of benign users
$\mathcal{X}_{\mathcal{N}(i)}$	Features of user i's neighborhood
$\hat{\eta}$	The reconstructed adjacency matrix generated at the decoder
γ	The Laplacian matrix based on the benign weights
$\hat{\gamma}$	The reconstructed Laplacian matrix regenerated by the malicious user
$\hat{\boldsymbol{\lambda}}$	The reconstructed feature matrix

III. DFL Formulation and Defense Model

In this section, we present a DFL training process with benign users, the UIP threat model, as well as state-of-the-

art cosine similarity-based and Euclidean distance-based defense models. Table II lists the notation used in the paper.

A. DFL with Benign Users

Consider a DFL training process involving N benign users, as shown in Fig. 1. Each benign user, labeled $n \in [1, N]$, possesses a dataset of size $\mathbb{D}_n(\mathcal{T})$ during the $\mathcal{T}$-th round. An individual data sample collected by user n is represented as $x(a_n) \in [1, \mathbb{D}_n(\mathcal{T})]$, where $\mathcal{T}$ ranges from 1 to T with T being the total number of training rounds in the DFL process. Let $y(a_n)$ indicate the model's output for the sample $x(a_n)$. The training loss function for user n, expressed as $\mathcal{E}(\boldsymbol{\omega}_n(\mathcal{T}); x(a_n), y(a_n))$, measures the error in approximating the relationship between the input a_n and its corresponding output $y(a_n)$, where $\boldsymbol{\omega}_n(\mathcal{T})$ denotes the local model parameters of user n. Note that $\boldsymbol{\omega}_n(\mathcal{T})$ contains the trainable variables of the neural network (or other machine learning model, e.g., SVM) that user n optimizes on its private datasets during local model training. After each local model update is generated, $\boldsymbol{\omega}_n(\mathcal{T})$ is exchanged with neighboring users for aggregation in DFL.

Given $\mathbb{D}_n(\mathcal{T})$, the loss function of the DFL in the $\mathcal{T}$-th round is defined as [22]

$$\delta(\boldsymbol{\omega}_n(\mathcal{T})) = \frac{1}{\mathbb{D}_n(\mathcal{T})} \sum_{x(a_n)=1}^{\mathbb{D}_n(\mathcal{T})} \mathcal{E}(\boldsymbol{\omega}_n(\mathcal{T}); x(a_n), y(a_n)) + \alpha \cdot \mathcal{N}(\boldsymbol{\omega}_n(\mathcal{T})), \quad (1)$$

where $\mathcal{N}(\cdot)$ is a regularizer function that represents the effect of the local training noise, and $\alpha \in [0, 1]$ is a coefficient.

At the $\mathcal{T}$-th round, user n generates $\boldsymbol{\omega}_n(\mathcal{T})$ while receiving model updates from its neighbors. Let $\tilde{n}$ and $\boldsymbol{\omega}_i^S(\mathcal{T})$ represent the number of user n's neighbors, and the model update shared by neighbor $i \in [1, \tilde{n}]$, respectively. Define a consensus coefficient $\varrho_{n,i}^{\mathcal{T}}$ that is used by user n to aggregate the models received from its neighbors [23]. In DFL, it is critical to achieve consistent $\boldsymbol{\omega}_n(\mathcal{T})$ across all users upon convergence, thus $\varrho_{n,i}^{\mathcal{T}}$ is designed to achieve the consistency [24]. $\varrho_{n,i}^{\mathcal{T}}$ can be expressed as [25]

$$\varrho_{n,i}^{\mathcal{T}} = \begin{cases} \zeta^{\mathcal{T}}, & \text{if } (n,i) \in \mathcal{C}^{\mathcal{T}}; \\ 1 - \tilde{n}\zeta^{\mathcal{T}}, & \text{if } n = i; \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where $\mathcal{C}$ represents the set of edges if user n and its neighbor are connected at the $\mathcal{T}$-th round. Given a topology in $\mathcal{T}$, $\zeta^{\mathcal{T}}$ is a constant achieved when $\boldsymbol{\omega}_n(\mathcal{T})$ are consistent across the neighbors, i.e., DFL converges at user n [25].

Based on (1), the optimal model update of user n can be given as

$$\boldsymbol{\omega}_n^*(\mathcal{T}) = \sum_{i \in [1,\tilde{n}]} \varrho_{n,i}^{\mathcal{T}} \frac{\mathbb{D}_i(\mathcal{T})}{\mathbb{D}(\mathcal{T})} \delta(\boldsymbol{\omega}_i^S(\mathcal{T})) + \frac{\mathbb{D}_n(\mathcal{T})}{\mathbb{D}(\mathcal{T})} \delta(\boldsymbol{\omega}_n(\mathcal{T})). \quad (3)$$

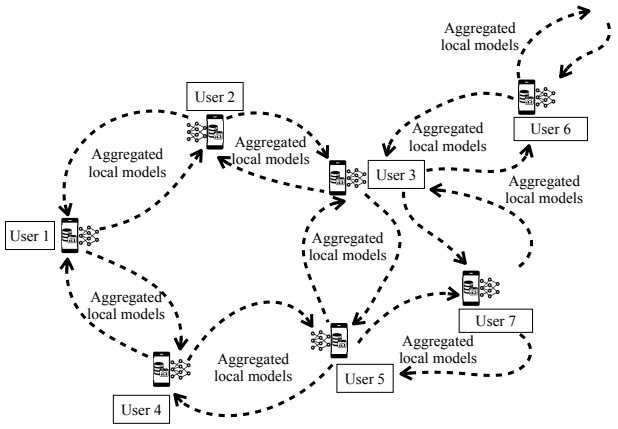


Fig. 1: A training process of the local and shared models in DFL, where an user n trains its datasets $\mathbb{D}_n(\mathcal{T})$ to generate a local model update $\boldsymbol{\omega}_n(\mathcal{T})$, and aggregate the shared models $\boldsymbol{\omega}_n^S(\mathcal{T})$ from the neighbors.

Then, a shared model update, denoted by $\boldsymbol{\omega}_n^S(\mathcal{T}) \leftarrow \boldsymbol{\omega}_n^*(\mathcal{T})$, can be created at user n and shared with its $\tilde{n}$ neighbors for the further training of $\boldsymbol{\omega}_n^S(\mathcal{T}+1)$, i.e.,

$$\boldsymbol{\omega}_n^S(\mathcal{T}+1) \leftarrow \boldsymbol{\omega}_n^S(\mathcal{T}) - \beta \cdot \nabla \delta(\boldsymbol{\omega}_n^S(\mathcal{T})), \quad (4)$$

where β is the learning rate of the users.

B. UIP Threat Model

Fig. 2 depicts the UIP threat model, where any malicious user j, $\forall j \in [1, N']$ generates its UIP model updates $\boldsymbol{\omega}_j^a(\mathcal{T})$ based on $\boldsymbol{\omega}_n^S(\mathcal{T})$ collected from its neighbors. The DFL contains N' authorized (legitimate) but malicious users [26], who attempt to progressively isolate the benign model features from DFL by creating and uploading malicious local models during each communication round.

Assume that any UIP model updates remain undetected during training. Although the benign users may not be aware of any attackers, it is prudent for them to be cautious about potential presence of malicious participants and their compromised models. The benign users are expected to continuously monitor and assess the shared models received from neighbors to detect any malicious model updates.

Note that the proposed UIP attack does not rely on intercepting or modifying communication channels. Instead, a malicious user is modeled as a legitimate but malicious participant in DFL. By design, DFL assumes that each user exchanges model updates with its neighbors in a peer-to-peer fashion; therefore, a compromised user has access to its own outgoing and incoming model updates. In this case, encryption protects against external attackers but cannot prevent such an insider adversary from manipulating its own updates. Even with encrypted transmissions, once the shared model updates are received and decrypted, a malicious user can apply the proposed threat model to craft malicious UIP model updates.

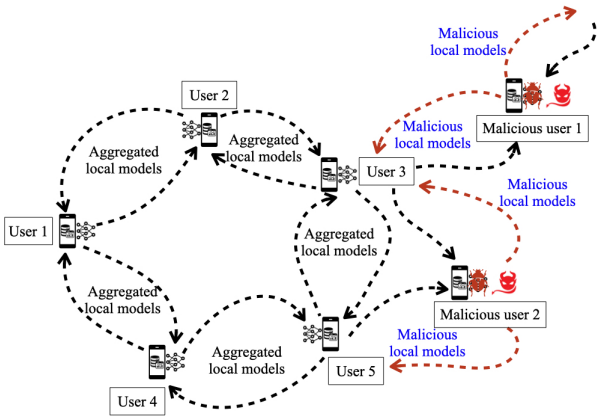


Fig. 2: For achieving UIP on DFL, the malicious model updates $\boldsymbol{\omega}_j^a(\mathcal{T})$ are tailored based on the collected $\boldsymbol{\omega}_n^S(\mathcal{T})$ from the neighbors. The proposed MPG framework at the adversarial users aims to maximize the DFL training loss, where the manipulated $\boldsymbol{\omega}_j^a(\mathcal{T})$ corrodes the benign one.

C. Defense Models

At user n, the cosine similarity among the received model updates from its $\tilde{n}$ neighbors can be measured, which is used as a defense mechanism [27], [28] to detect potential UIP model updates.

The cosine similarity calculates the angular similarity between every two users' model updates, as given by

$$\overline{\omega}_{i',i''} = \frac{\boldsymbol{\omega}_{i'}(\mathcal{T}) \cdot \boldsymbol{\omega}_{i''}(\mathcal{T})}{\|\boldsymbol{\omega}_{i'}(\mathcal{T})\| \cdot \|\boldsymbol{\omega}_{i''}(\mathcal{T})\|}, \tag{5}$$

where $(i', i'') \in [1, \tilde{n}]$, $i' \neq i''$, and $\|\cdot\|$ stands for length (magnitude) of the vector.

By computing the cosine similarity for each neighbor's model update, user n aims to identify those model updates that deviate significantly in direction from the others. If the similarity is beyond a predetermined threshold, denoted by d_T^{Cos}, the update can be flagged as potentially malicious. This approach relies on the assumption that malicious model updates deviate significantly in direction from benign ones, enabling their detection and allowing the aggregation process to discard or down-weight them accordingly.

We also consider another typical popular attacks' detection model residing at the server, which leverages the Euclidean distance metric to discern malicious local models, for instance, Krum [29] or Multi-Krum [30]. By measuring the Euclidean distance between each received local model and the aggregated model, this model aims to identify anomalous deviations indicative of malicious intent. The underlying rationale is that genuine local models from benign devices are expected to cluster within a certain proximity in the model space, while malicious local models, designed to sabotage the shared model's integrity, would exhibit more pronounced deviations. By setting a distance threshold, denoted by d_T^{Euc}, local models that exceed this threshold can be flagged or discarded, effectively isolating and mitigating the impact of malicious local models on the shared model's integrity.

IV. Proposed MPG-based UIP attack on DFL

In this section, we delineate the architecture of the MPG that aims to generate malicious model updates tailored for UIP. The graph signal processing based on an adversarial graph autoencoder (AGAE) is developed within the MPG, and trained together with sub-gradient descent to reconstruct manipulatively the correlations of the model updates, where the reconstruction loss is maximized.

A. Adversarial Message-Passing Neural Networks

At malicious user j, $\boldsymbol{\omega}_j^a(\mathcal{T})$ can be optimized to maximize the loss function of DFL in (3), thereby isolating the benign user's contributions and skewing the shared model. Given the aggregation function of the DFL in (3) and (5), the optimization of the adversarial training model at the malicious user j for isolating a benign user n can be formulated as

$$\max_{\boldsymbol{\omega}_j^a(\mathcal{T})} \Big(\sum_{i\in[1,\tilde{n}']} \varrho_{n,i}^{\mathcal{T}} \frac{\mathbb{D}_i(\mathcal{T})}{\mathbb{D}(\mathcal{T})} \delta(\boldsymbol{\omega}_i^S(\mathcal{T})) + \frac{\mathbb{D}_n(\mathcal{T})}{\mathbb{D}(\mathcal{T})} \delta(\boldsymbol{\omega}_n(\mathcal{T})) + \sum_{j\in[1,N']} \varrho_{n,j}^{\mathcal{T}} \frac{\mathbb{D}_j(\mathcal{T})}{\mathbb{D}(\mathcal{T})} \delta(\boldsymbol{\omega}_j^a(\mathcal{T})) \Big) \tag{6a}$$

$$\text{s.t.} \quad \overline{\omega}_{i',i''} \leq d_T^{\text{Cos}}, \tag{6b}$$

$$d(\delta(\boldsymbol{\omega}_{i'}^S(\mathcal{T}), \delta(\boldsymbol{\omega}_{i''}^S(\mathcal{T})) \leq d_T^{\text{Euc}}, \tag{6c}$$

where $\tilde{n}' = \tilde{n} - N'$, and $d(\delta(\boldsymbol{\omega}_{i'}^S(\mathcal{T}), \delta(\boldsymbol{\omega}_{i''}^S(\mathcal{T}))$ in (6c) evaluates the Euclidean distance between any two shared model updates received at benign user n. As the malicious user participates as a legitimate user, the thresholds d_T^{Cos} and d_T^{Euc} are known to all users in DFL.

Constraints (6b) and (6c) guarantee that the attacker's malicious local model $\boldsymbol{\omega}_j^a(\mathcal{T})$ is in proximity to the shared model in terms of cosine similarity and Euclidean distance, while the two constraints ensure the overall similarity and distance between the selected local models and their shared model is below the upper bounds d_T^{Cos} and d_T^{Euc}, respectively. The A-MPNN is tailored to ensure those constraints and hinder the convergence of DFL by isolating the correlation features between the benign local models and embedding the correlation features in graphs for malicious UIP model generation.

Note that the optimization in (6) is not an attempt to obtain a provably convergent learner for DFL but rather the formulation of an adversarial objective: it presents the UIP attack to maximize the DFL training loss so as to restrain or prevent DFL convergence. To solve (6) in the UIP threat model, a new adversarial message-passing neural networks (A-MPNN) architecture is developed at the malicious user, as shown in Fig. 3. Specifically, a graph G can be constructed, where the received models $\boldsymbol{\omega}_n^S(\mathcal{T})$ are represented as nodes.

Let $\tilde{j}$ denote the number of benign neighbors around malicious user j, which is also the number of nodes in G

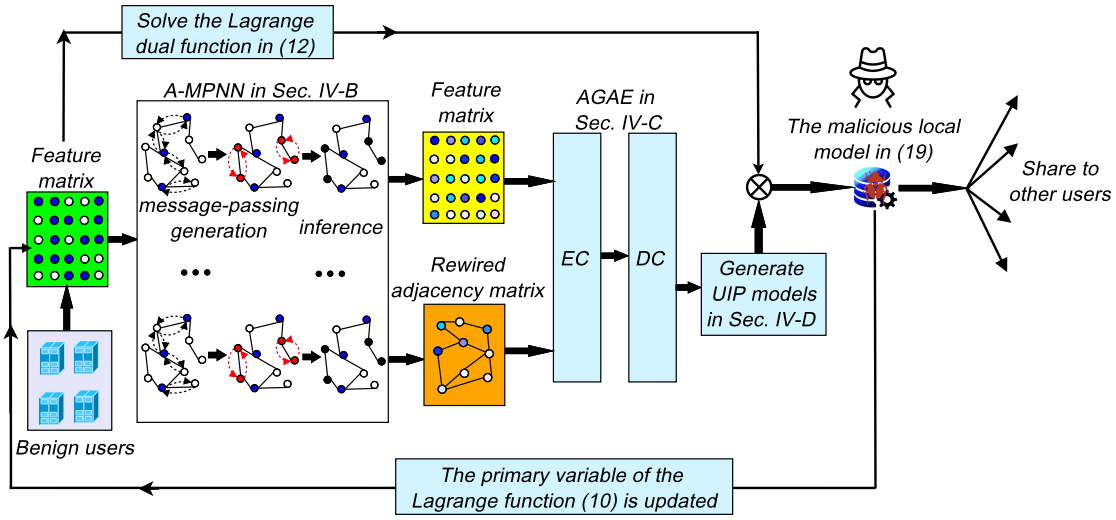


Fig. 3: With $\boldsymbol{\omega}_n^S(\mathcal{T})$ from the neighbors, the proposed MPG-based UIP attack enables the malicious user to obtain the hidden representations of each feature in the graph data. According to the model correlation, MPG is designed to generate the UIP model update $\boldsymbol{\omega}_j^a(\mathcal{T})$ that aims to isolate $\boldsymbol{\omega}_n(\mathcal{T})$ of the benign users.

$(n \in [1, \tilde{j}])$. A matrix that contains $\boldsymbol{\omega}_n^S(\mathcal{T})$ is an input to the adversarial message-passing layer, while connectivity is encoded in an adjacency matrix $\eta \in \mathbb{R}^{\tilde{j}\times\tilde{j}}$. An encoder of the adversarial message-passing layer, denoted by h, uses message-passing operations over $\boldsymbol{\omega}_n^S(\mathcal{T})$ to produce neighborhood-aware representations of the context set.

Given τ message passing steps, the features of the node $\boldsymbol{\omega}_n^S(\mathcal{T})$ are denoted by λ_n^τ, and the edges of two nodes n and n' on graph G are denoted by $\mu_{n,n'}^\tau$, where $\eta_{n,n'}^\tau = 1$. To update the node features, the message passing generation function of the proposed MPG can be written as [31]

$$\lambda_n^{\tau+1} \triangleq F(\lambda_n^\tau, \Psi_{n'\in\mathcal{N}(n)}, \mathcal{G}(\lambda_n^\tau, \lambda_{n'}^\tau, \mu_{n,n'}^\tau)), \quad (7)$$

where F and $\mathcal{G}$ are learnable functions, Ψ is a permutation-invariant aggregation function, and $\mathcal{N}(n) = \{n' | \eta_{n,n'}^\tau = 1\}$.

Let $\mathcal{X}_{\mathcal{N}(i)}$ denote the features of node i's neighborhood on graph G. According to [32] and (7), the adversarial message-passing layer in MPG can be written as

$$\Pr(\mathcal{Z}, \mathcal{Y}_{1:i} | \mathcal{X}_{1:i}, \bigcup_{i=1}^{\tilde{n}'} \mathcal{X}_{\mathcal{N}(i)}) = \Pr(\mathcal{Z}) \Pi_{i=1}^{\tilde{n}'} \cdot \mathcal{N}(\mathcal{Y}_i |$$
$$F(\mathcal{X}_i || \mathcal{Z}, \Psi_{n'\in\mathcal{N}(i)}, \mathcal{G}(\mathcal{X}_i || \mathcal{Z}, \mathcal{X}_{n'} || \mathcal{Z})), \sigma^2), \quad (8)$$

where $\mathcal{Z}$ is a random vector mimicking the randomness of message passing, and σ is a random distribution of observation noise [33].

Through iterative message passing, the A-MPNN aggregates neighboring benign models to learn latent feature correlations. Instead of faithfully representing these correlations, the adversarial component modifies them to strategically suppress benign users' influence during aggregation. This allows the attacker to craft malicious updates that outwardly resemble benign ones yet progressively diminish the contributions of benign users to the shared model.

The significance of the A-MPNN lies in its ability to manipulate complex graph-level dependencies that cannot be detected by conventional distance/similariy-based defenses. While many existing methods aim at spotting anomalies in weight space using the Euclidean distance or cosine similarity, the A-MPNN exploits higher-order correlations among model updates to gradually bias DFL. In decentralized settings, the absence of a central server makes it difficult to monitor or filter such manipulations. The A-MPNN leads to the unique risks of DFL and the limitations of defense strategies. Beyond demonstrating a new attack, the A-MPNN also points to the necessity of graph-aware defenses that can reason about structural dependencies rather than relying solely on parameter similarity.

B. Training Adversarial MPG with Sub-Gradient Descent

Optimizing the adversarial training model at the malicious user, as specified in (6), presents a non-convex combinatorial problem that is intractable with standard optimization techniques. To address this, we extend the Lagrangian dual method to decouple the MPG architecture, effectively separating the attack mechanism from the selection of benign users. We design a new iterative approach, as illustrated in Fig. 3, to optimize the UIP model updates $\boldsymbol{\omega}_j^a(\mathcal{T})$ by concurrently running the adversarial MPG and updating sub-gradient descent.

According to (6a), (6b), and (6c), let $\rho(\mathcal{T})$ and $\theta(\mathcal{T})$ denote the dual variables, and the Lagrange function at the malicious user j can be written as

$$\mathcal{L}_j(\rho(\mathcal{T}), \theta(\mathcal{T})) = \mathcal{F}_{\text{Loss}}(\boldsymbol{\omega}_j^a(\mathcal{T})) + \sum_{i',i''=1}^{\tilde{n}} \rho(\mathcal{T})(d_T^{\text{Cos}}$$
$$- \overline{\omega}_{i',i''}) + \sum_{i',i''=1}^{\tilde{n}} \theta(\mathcal{T})(d_T^{\text{Euc}} - d(\delta(\boldsymbol{\omega}_{i'}^S(\mathcal{T}), \delta(\boldsymbol{\omega}_{i''}^S(\mathcal{T}))))), \quad (9)$$

where $i' \neq i''$, and $\mathcal{F}_{\text{Loss}}(\boldsymbol{\omega}_j^a(\mathcal{T}))$ represents the objective function in (6a).

We further rewrite the Lagrange dual function as

$$\mathcal{L}_j^{\mathcal{D}}(\rho(\mathcal{T}),\theta(\mathcal{T})) = \max_{\boldsymbol{\omega}_j^a(\mathcal{T})} \mathcal{L}_j(\rho(\mathcal{T}),\theta(\mathcal{T})). \tag{10}$$

The dual problem of the problem in (6) can be given by

$$\min_{\rho(\mathcal{T}),\theta(\mathcal{T})} \mathcal{L}_j^{\mathcal{D}}(\rho(\mathcal{T}),\theta(\mathcal{T})). \tag{11}$$

At the malicious user j, the primary variable $\boldsymbol{\omega}_j^a(\mathcal{T})$ of the Lagrange function (10) can be updated by solving

$$\boldsymbol{\omega}_j^a(\mathcal{T})^* = \arg\max_{\boldsymbol{\omega}_j^a(\mathcal{T})} \Big\{ \mathcal{F}_{\mathrm{Loss}}(\boldsymbol{\omega}_j^a(\mathcal{T})) - \sum_{i',i''=1}^{\tilde{n}} \rho(\mathcal{T})(d_T^{\mathrm{Cos}} - \overline{\omega}_{i',i''}) - \sum_{i',i''=1}^{\tilde{n}} \theta(\mathcal{T}) \Big(d_T^{\mathrm{Euc}} - d\big(\delta(\boldsymbol{\omega}_{i'}^S(\mathcal{T}), \delta(\boldsymbol{\omega}_{i''}^S(\mathcal{T}))\big)\Big)\Big\}. \tag{12}$$

To optimize the adversarial model parameters $\boldsymbol{\omega}_j^a(\mathcal{T})^*$ in (12), we design a new AGAE within the proposed MPG architecture. As illustrated in Fig. 3, the AGAE consists of two primary components: An encoder and a decoder. The encoder is designed to encode the feature matrix λ_n by utilizing the rewired adjacency matrix η from the MPG, which outputs a matrix $\mathbf{Z}$.

Let $F_{\mathrm{cov}}(\cdot|\cdot)$, $\mathbf{w}$, and $\mathcal{B}$ denote a spectral convolution function, a weight matrix, and the number of graph convolutional network's layers, respectively, we have $\mathbf{Z}^{\mathcal{B}} = F_{\mathrm{cov}}(\mathbf{Z}^{\mathcal{B}-1}, \eta|\mathbf{w}^{\mathcal{B}})$ [34]. Let $\mathcal{F}_G(\mathbf{Z}^{\mathcal{B}-1}, \eta|\mathbf{w}^{\mathcal{B}})$ represent the graph G, and $\Phi^{\mathcal{B}}(\cdot)$ denote a nonlinear activation function, e.g., tanh or ReLU. The encoder can be given by

$$\mathcal{F}_G(\mathbf{Z}^{\mathcal{B}-1}, \eta|\mathbf{w}^{\mathcal{B}}) = \Phi^{\mathcal{B}}(\overline{\eta}^{-\frac{1}{2}}\widetilde{\eta}\overline{\eta}^{-\frac{1}{2}}\mathbf{Z}^{\mathcal{B}-1}\mathbf{w}^{\mathcal{B}}), \tag{13}$$

where $\widetilde{\eta}=\eta+\mathcal{I}$, $\mathcal{I}$ denotes an identity matrix, and $\overline{\eta}=\sum\widetilde{\eta}$.

The decoder in AGAE then takes the output $\mathbf{Z}^{\mathcal{B}}$ from the encoder to reconstruct a UIP $\hat{\eta}$. This operation essentially acts as the inverse of the encoder. The decoder aims to reconstruct the original graph from its compressed representation produced by the encoder. We define

$$\hat{\eta} = \mathrm{Sigmoid}(\mathbf{Z}^{\mathcal{B}}(\mathbf{Z}^{\mathcal{B}})^{\mathrm{T}}), \tag{14}$$

where $\mathrm{Sigmoid}(x) = 1/(1+e^{-x})$ is the Sigmoid function.

To evaluate the reconstruction accuracy, the decoder's output is compared against the original input graph, and a loss function is formulated based on the differences. The encoder and decoder are trained simultaneously in an end-to-end manner to maximize the reconstruction loss, thereby crafting a UIP $\boldsymbol{\omega}_j^a(\mathcal{T})^*$ from the reconstructed graph.

The reconstruction loss can be written as $\mathbb{E}_{\mathcal{F}_G(\mathbf{Z}^{\mathcal{B}-1},\eta|\mathbf{w}^{\mathcal{B}})}\Big[\log \vartheta(\ \hat{\eta}\ |\ \mathbf{Z}^{\mathcal{B}}\ )\Big]$, where $\vartheta(\ \hat{\eta}\ |\ \mathbf{Z}^{\mathcal{B}}\ ) = \Pi_{n=1}^{\tilde{j}}\Pi_{n'=1}^{\tilde{j}}\vartheta(\ \hat{\eta}\ |\ \mathbf{Z}_n^{\mathcal{B}}, \mathbf{Z}_{n'}^{\mathcal{B}}\ )$ and $\vartheta(\hat{\eta}=1|\mathbf{Z}_n^{\mathcal{B}}, \mathbf{Z}_{n'}^{\mathcal{B}}\ ) = \mathrm{sigmoid}\left(\mathbf{Z}_n^{\mathcal{B}}\left(\mathbf{Z}_{n'}^{\mathcal{B}}\right)^{\mathrm{T}}\right)$.

The proposed AGAE uses reconstruction loss as a proxy for how well the learned latent graph captures the genuine feature-level correlations among benign model updates. By maximizing this loss, the adversary forces the latent representation to misrepresent those correlations. During message passing, the AGAE learns node embeddings that allow reconstruction of neighborhood feature patterns; the adversary optimizes its UIP model updates to increase the reconstruction error for benign nodes. As a result, (i) the embedding geometry is perturbed so that benign users no longer lie in the same latent neighborhoods as before, and (ii) the pairwise affinity/attention coefficients (computed from embeddings) are altered so that incoming messages from benign neighbors are down-weighted in aggregations.

C. Generating UIP Models

The proposed MPG-based UIP attack represents a new category of adversarial threats in DFL, where malicious users exploit feature-level correlations in local model updates to strategically isolate benign participants' contributions. Specifically, the MPG learns the feature-level correlations across neighboring model updates through iterative message passing, and then modifies these correlations to craft malicious updates that reduce the contribution of benign peers in subsequent aggregations. Through message passing, the adversarial MPG captures local dependencies among benign users' updates and selectively alters them to weaken benign nodes' influence in the evolving model graph. Over multiple rounds, this controlled injection of adversarial signals results in the progressive isolation of benign participants, causing their updates to be marginalized in the global consensus.

According to Fig. 3, Algorithm 1 is executed at the malicious user j to train the proposed MPG, where the output of the AGAE refines the the correlation of $\boldsymbol{\omega}_j^a(\mathcal{T})$ with the benign ones. To generate $\boldsymbol{\omega}_j^a(\mathcal{T})$, we define a Laplacian matrix based on η, which yields

$$\gamma = \Delta(\eta) - \eta, \tag{15}$$

where $\Delta(\eta)$ denotes a diagonal matrix of η.

Let Λ_{GFT} denote a graph Fourier transform (GFT) basis, which can transform graph data to its spectral-domain representation. $\Delta'(\eta)$ presents a diagonal matrix with the eigenvalues of η along its main diagonal. Singular value decomposition (SVD) can be applied to (15), yielding

$$\gamma = \Lambda_{\mathrm{GFT}}\Delta'(\eta)\Lambda_{\mathrm{GFT}}^{\mathrm{T}}. \tag{16}$$

First removing inter-model correlations and then emphasizing the underlying data features supporting the local models, we construct a matrix $\mathcal{H}$ that represents the spectral-domain data characteristics of the benign local models.

Given the feature matrix $\boldsymbol{\lambda}$ that contains the features of neighboring benign local models, the malicious user has

$$\mathcal{H} = \Lambda_{\mathrm{GFT}}^{-1}\boldsymbol{\lambda}. \tag{17}$$

Similar to (15), with $\hat{\eta}$ from the decoder of AGAE, the malicious user can build a reconstructed Laplacian matrix:

$$\hat{\gamma} = \Delta(\hat{\eta}) - \hat{\eta}. \tag{18}$$

The SVD of $\hat{\gamma}$ determines the reconstructed GFT basis, $\overline{\Lambda}_{\mathrm{GFT}}$. According to (17), the malicious user can obtain a reconstructed feature matrix, as given by

$$\hat{\boldsymbol{\lambda}} = \overline{\Lambda}_{\mathrm{GFT}}\mathcal{H}. \tag{19}$$

Algorithm 1 Training MPG for Generating UIP Models

1: The malicious user j, $\forall j \in [1, N']$ collects $\boldsymbol{\omega}_n^S(\mathcal{T})$ from its neighbors.
2: Given τ message passing steps, graph G is formulated at each malicious user with λ_n^τ and $\mu_{n,n'}^\tau$.
3: The adjacency matrix η^τ is initialized with G.
4: **Training the A-MPNN**:
5: **for** step = 1, 2, ..., τ **do**
6: λ_n^τ is updated by (7) while $\mu_{n,n'}^\tau$ is rewired according to the update of (8).
7: **end for**
8: **Training MPG with sub-gradient descent**:
9: λ_n and $\mu_{n,n'} \rightarrow$ AGAE, where λ_n is encoded $\rightarrow \mathbf{Z}^{\mathcal{B}}$.
10: At the decoder, $\hat{\eta}$ is reconstructed by (14), while $\mathbb{E}_{\mathcal{F}_G(\mathbf{Z}^{\mathcal{B}-1}, G|\mathbf{w}^{\mathcal{B}})}\Big[\log\ \vartheta(\ \hat{\eta}\ |\ \mathbf{Z}^{\mathcal{B}}\ )\Big]$ is maximized.

In particular, $\hat{\boldsymbol{\lambda}}$ determines the malicious local model $\boldsymbol{\omega}_j^a(\mathcal{T})$ whose correlation with the benign local models is refined by AGAE to achieve the optimization of the adversarial training model in (6).

D. Complexity Analysis

Each message-passing step in the A-MPNN involves aggregating features from neighbors and updating node embeddings. For a graph with N benign users and $\mathcal{C}$ edges, the computational complexity per message-passing step is $\mathcal{O}(|\mathcal{C}|\kappa)$, where κ is the embedding dimension. In sparse DFL, where each user connects to a limited set of neighbors and thus $|E| \approx \mathcal{O}(N)$, the per-step cost becomes linear in the number of users. With τ message-passing steps and $\mathcal{B}$ layers, the overall complexity is $\mathcal{O}(\tau\mathcal{B}N\kappa)$, which remains scalable to hundreds or even thousands of users given that both τ and $\mathcal{B}$ are modest constants in our design. Memory requirements are also linear in N, as each user maintains an embedding vector and neighbor indices.

V. Performance Evaluation

This section presents the implementation of the MPG-based UIP attack using PyTorch. We assess the test accuracy based on $\delta(\boldsymbol{\omega}_n(\mathcal{T}))$, as well as the KL divergence among the received shared models, when subjected to the MPG-based UIP attack. The detection efficacy of the MPG-based UIP attack is examined through the metric of cosine similarity and Euclidean distance between $\boldsymbol{\omega}_n(\mathcal{T})$ and $\boldsymbol{\omega}_i^S(\mathcal{T})$, $\forall i \in [1, \tilde{n}]$. The source code for the MPG-based UIP attack has been released on GitHub: https://github.com/AnonymousAuthors/DFL_Attack.

A. Experimental Implementation

The standard DFL typically seeks to enhance accuracy in image classification; conversely, the UIP attack described here intentionally deteriorates performance and provokes incorrect labeling. In our setup, the number of benign users (N) is set to 40, whereas the malicious users (N') are given as 5 or 10. The model aggregation of $\boldsymbol{\omega}_n^S(\mathcal{T})$, undergoes training across 50 communication rounds, with each local model $\boldsymbol{\omega}_n(\mathcal{T})$ performing 10 iterations per round.

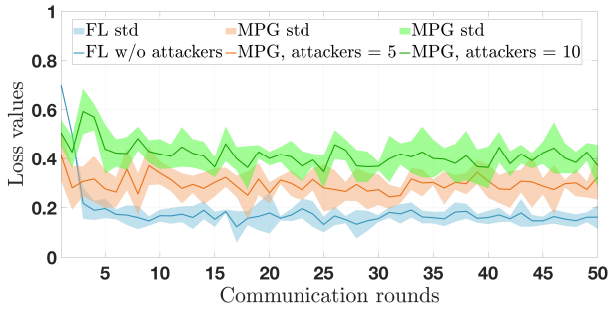
For constructing the adjacency matrix η at A-MPNN, we experiment by selecting different numbers of message passing steps (τ), specifically 100, 200, and 300, so that $\mu_{n,n'}^\tau$ is rewired according to the update of (8). The AGAE's encoder consists of a two-layer Graph Convolutional Network (GCN) enhanced by a dropout layer for mitigating overfitting risks, while the decoder employs an inner-product operation. Optimization is carried out using the Adam optimizer at a learning rate of 0.01.

The implementation of the proposed MPG attack was conducted on a SVM model, utilizing PyTorch version 1.12.1 and Python version 3.9.12. This setup was deployed on a Linux-based workstation, equipped with an Intel(R) Core(TM) i7-9700K CPU at 3.60 GHz, featuring 8 cores, and supported by 16 GB of DDR4 memory operating at 2400 MHz. The experimentation involved the application of the UIP attack across three datasets, demonstrating the attack's efficacy and potential impacts on SVM models under specified computational environments and data conditions:

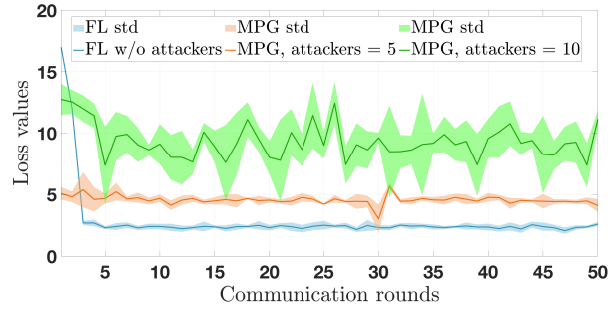
- MNIST: This is a common basic dataset containing grayscale images of handwritten digits ranging from 0 to 9, split into a training set of 60,000 samples and a test set of 10,000 samples.
- CIFAR-10: This dataset includes 60,000 color images sized 32 $\times$ 32 pixels, categorized into 10 classes, each with 6,000 images. This dataset divides into 50,000 training images and 10,000 test images.
- Street View House Numbers (SVHN): This dataset comprises more than 600,000 digit images extracted from real-world street-view captures, representing house numbers in natural, unprocessed scenes, displaying significant variability in illumination, viewing angle, and background context.

Note that, regardless of the specific model architecture (NNs or SVMs) employed for model training at benign users, the fundamental premise of the UIP attack is valid. Specifically, the attack does not depend on the specific architecture used by benign users, but rather on the exchange of shared model updates and their feature-level correlations. In practice, whether the shared model is trained using SVMs or NNs, the adversary can still generate UIP model updates that, when aggregated, steer the DFL away from its optimal shared models. This is achieved through the new A-MPNN, which is designed to capture the structural correlations of benign users' model updates and then craft malicious model updates that effectively isolate benign contributions. Because these correlations are inherent to the shared model regardless of whether it is built on NNs or SVMs, the UIP attack is a general and effective threat across different model architectures in DFL.

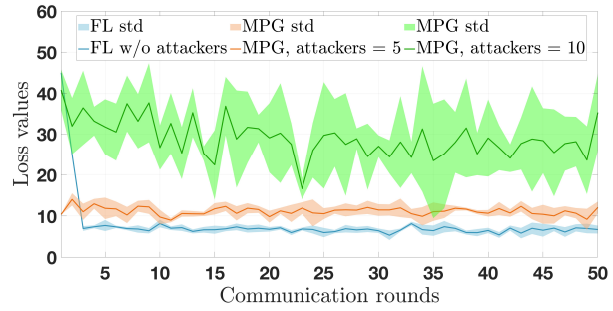
Recent studies [23], [35], [36] indicate that network topologies, such as ring and grid structures, can significantly impact DFL performance by influencing model propagation, thereby affecting model accuracy. Therefore, we examine these two topologies within our DFL framework.



(a) MNIST



(b) CIFAR-10



(c) SVHN

Fig. 4: The local model training loss of the proposed MPG attack on the MNIST, CIFAR-10, and SVHN datasets, where there are 50 communication rounds in the ring topology.

- Ring: each user is connected only to two immediate neighbors, forming a closed-loop structure. Model aggregation occurs locally, where each user combines its own model updates with those received from its immediate neighbors, propagating information sequentially around the ring.
- Grid: users are arranged in a two-dimensional lattice, where each user connects directly with its immediate horizontal and vertical neighbors. Aggregation at each user involves combining local model updates with models received from the adjacent neighbors, enabling parallel and structured information flow.

For performance comparison, we consider the following two existing, recently developed attack models as baselines.

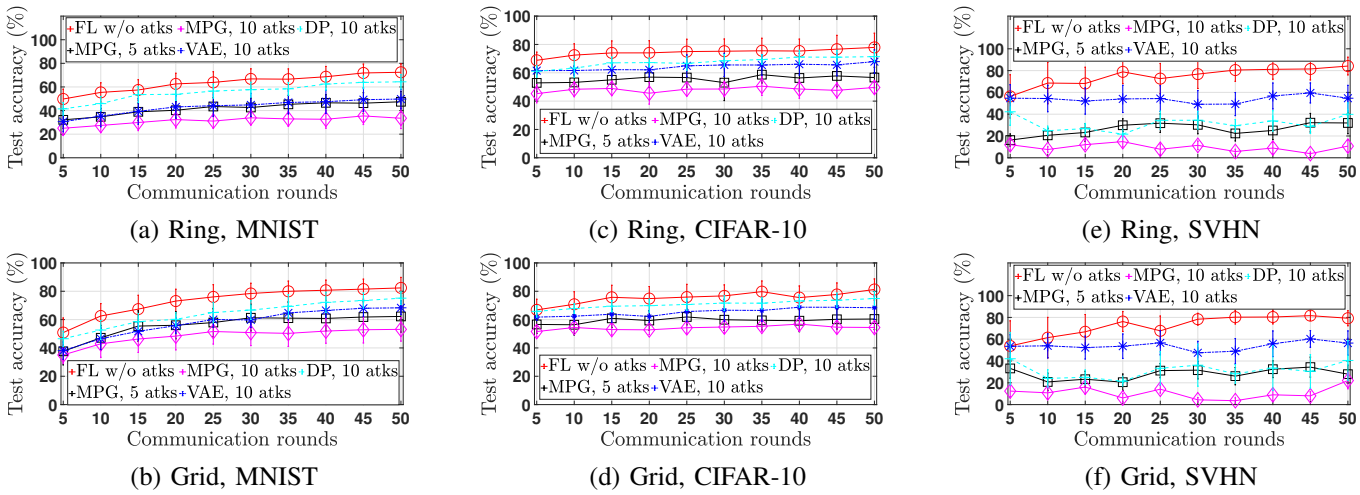
- Variational autoencoder (VAE)-based poisoning attack [21]: This attack model explores VAE to regenerate the graph's structural correlations adversarially to maximize the training loss of DFL, where adversarial graph structure alongside the benign training data features are used to create malicious local models. This approach assumes that the malicious users can passively intercept the local models shared by benign users and utilize them to generate the malicious local models. In particular, the VAE-based poisoning attack relies on an encoder-decoder framework to reconstruct model updates from latent variables. It does not incorporate graph message passing; in other words, it primarily captures global statistical dependencies while overlooking fine-grained, local correlations among users' updates.
- Differential privacy (DP)-based attack [37]: This attack model operates by injecting Gaussian noise into the malicious local model updates before sharing them with other users in DFL. The malicious users adjust the variance of this Gaussian noise over time to maximize disruption to the convergence and degrade overall model accuracy.

B. Attacking Performance

1) Training loss: Fig. 4 presents the local model's training loss under the proposed MPG attack given the MNIST, CIFAR-10, and SVHN datasets. With five malicious users participating in DFL, the average loss increases from 0.2 to 0.3 (MNIST), from 3 to 5 (CIFAR-10), and from 8 to 11 (SVHN) compared to DFL without attackers. Specifically, for the CIFAR-10 and SVHN datasets, further increasing the number of malicious users to ten results in a significant rise in the average loss values, from 5 to 13 for CIFAR-10 and from 11 to 31 for SVHN. This notable degradation demonstrates the effectiveness of our proposed UIP attack. By isolating benign users and forcing them to train on manipulated or low-quality data, our attack effectively amplifies the local training errors, which shows the vulnerability of DFL systems to user isolation strategies, particularly as the number of malicious users grows.

Moreover, the greater variability observed in Figs. 4(b) and 4(c) for CIFAR-10 and SVHN, compared to Fig. 4(a) for MNIST, is primarily due to the higher complexity and variability inherent in these datasets. Particularly, MNIST comprises simple, grayscale handwritten digits with limited variance, making the dataset less sensitive to perturbations introduced by the UIP attack. In contrast, CIFAR-10 and SVHN datasets consist of more complex, colored images with diverse backgrounds and higher intra-class variations, rendering the training process inherently more fluctuating. Under the UIP attack, the impact of maliciously injected models is significantly magnified, causing substantial disruption and resulting in a pronounced increase in the training loss for CIFAR-10 and SVHN compared to MNIST.

2) Network topologies: Fig. 5 compares the model test accuracy under our proposed MPG-based UIP attack against existing DP-based and VAE-based attacks, evaluated on both ring and grid topologies in the DFL framework. Specifically, the MPG-based UIP attack achieves



(a) Ring, MNIST (c) Ring, CIFAR-10 (e) Ring, SVHN

(b) Grid, MNIST (d) Grid, CIFAR-10 (f) Grid, SVHN

Fig. 5: Model test accuracy of the proposed MPG-based UIP attack and the existing DP-based or VAE-based attack on the MNIST, CIFAR-10, and SVHN datasets. Ring and grid topologies are considered.

a lower test accuracy in the ring topology compared to the grid topology. For instance, Figs. 5(a) and 5(b) show that, when deploying the MPG-based UIP attack with ten malicious users on the MNIST dataset, the test accuracy on the ring topology is 21.3% lower than on the grid. Similarly, Figs. 5(c) and 5(d) show a 5.7% accuracy reduction on CIFAR-10. Figs. 5(e) and 5(f) indicate an 11.7% accuracy drop on SVHN, when comparing ring topology performance to that of the grid topology.

The difference in test accuracy between the ring and grid topologies under our MPG-based UIP attack is reasonable. The ring topology provides limited connectivity between the users, where each user exchanges local models only with its immediate neighbors, resulting in longer and less redundant communication paths for propagating model updates across the DFL. Any malicious manipulation in such a sparsely connected structure is amplified, as benign users have fewer alternative trustworthy sources to mitigate the influence of UIP updates. Conversely, the grid topology features richer and more redundant connectivity, enabling multiple pathways for information propagation. This redundancy facilitates more effective filtering of malicious information, as each benign user aggregates updates from several neighbors, thus reducing the overall impact of any single compromised model update. The proposed MPG-based attack achieves greater performance degradation in the ring topology compared to the grid topology.

Moreover, the MPG-based UIP attack consistently outperforms existing DP-based and VAE-based attacks. As shown in Figs. 5(e) and 5(f), the test accuracy under the MPG-based attack is 28.8% lower than the DP-based attack, and 49.5% lower than the VAE-based attack.

On the one hand, the superior performance can be attributed to the strength of the proposed A-MPNN architecture, which effectively preserves strong feature correlations between the crafted malicious models and the benign ones. By accurately capturing and exploiting these correlations, the malicious models appear more plausible and trustworthy to benign users. The poisoned updates propagate more effectively within the DFL, which significantly degrades the overall accuracy.

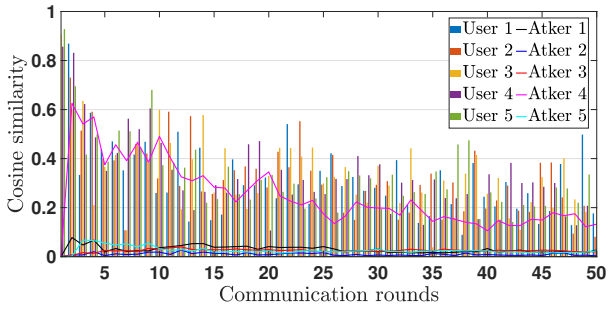


Fig. 6: Cosine similarity of the model updates under the proposed MPG attack.

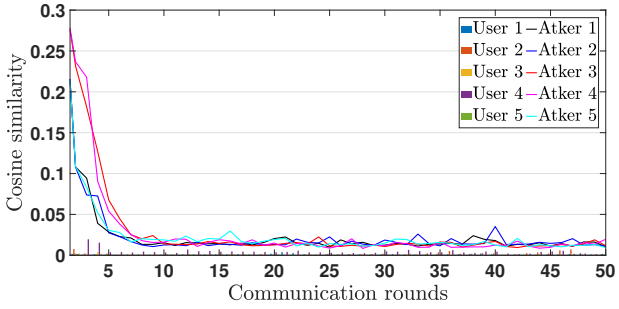


Fig. 7: Cosine similarity of the model updates under the VAE-based poisoning attack.

On the other hand, the DP-based attack inject random perturbations into model parameters to distort the training process; however, this randomness typically lacks coherence with the true underlying feature distributions, making the poisoned updates less effective in misleading

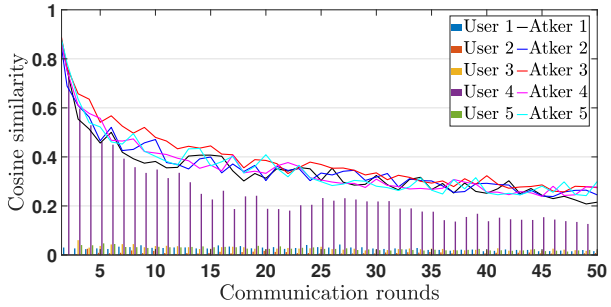


Fig. 8: Cosine similarity of the model updates under the DP attack.

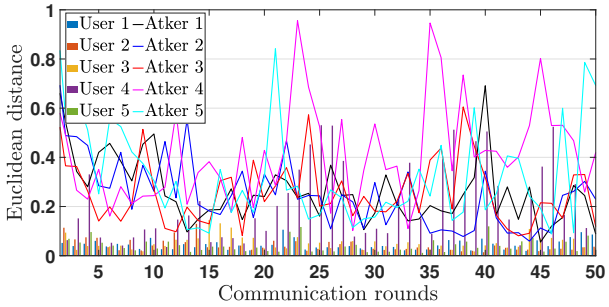


Fig. 10: Euclidean distance of the model updates under the VAE-based poisoning attack.

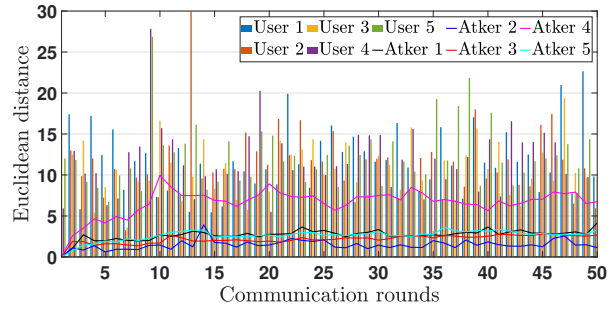


Fig. 9: Euclidean distance of the model updates under the proposed MPG attack.

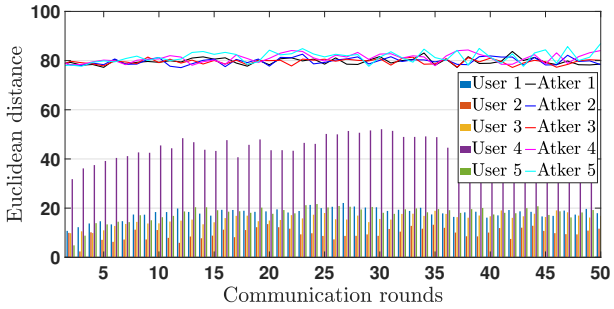


Fig. 11: Euclidean distance of the model updates under the DP attack.

benign models. In addition, the VAE-based attack rely on generative models trained to approximate data distributions, yet their reconstruction capabilities are constrained by the quality and diversity of latent representations learned during training. Therefore, the VAE-generated poisoned updates may miss crucial subtle correlations or hidden patterns that genuine models naturally exhibit.

3) Cosine similarity & Euclidean distance: To evaluate the stealthiness of the proposed UIP attack and compare it with existing poisoning strategies, Figs. 6–11 present the cosine similarity, as defined in (5), and the Euclidean distance, i.e., $d\big(\delta(\boldsymbol{\omega}_{i'}^{S}(\mathcal{T})), \delta(\boldsymbol{\omega}_{i''}^{S}(\mathcal{T}))\big)$, between the model updates of benign and malicious users under each attack. The evaluation is conducted in a DFL setting with a ring topology, where the total number of benign users N and the number of malicious users N' are set to 5.

The cosine similarities and Euclidean distances under the proposed MPG-based attack remain closer to those of benign local models compared to the DP- and VAE-based attacks. This indicates that the UIP model updates generated by the MPG attack are more similar to benign updates, making it significantly harder for the DFL users to distinguish and defend against manipulation. In contrast, the DP and VAE attacks result in much larger deviations, causing the malicious updates to stand out and become more easily detectable. The key strength of the proposed A-MPNN lies in its ability to selectively isolate a targeted benign user's feature representation while maintaining a high degree of correlation with the remaining benign model features. This targeted isolation allows the attack to degrade the model performance for the victim user without triggering anomaly detection mechanisms that rely on the neighborhood similarity metrics.

4) Ablation study: The VAE-based attack serves as an ablation baseline for our proposed MPG-based attack to highlight the importance of A-MPNN. Architecturally, the two threat approaches utilize graph neural networks to generate malicious model updates by learning from benign models. However, the MPG-based framework integrates message passing layers into the encoder-decoder structure, enabling the model to explicitly capture relational dependencies among neighboring benign users in the DFL topology.

As shown in Fig. 5, these message passing layers allow the MPG attack to encode topological and feature-level interactions between the aggregated local models at the malicious user, resulting in more context-aware and strategically crafted UIP model updates to isolate the benign features. In contrast, the VAE-based attack lacks this relational modeling capability, as it operates solely on individual model parameters without accounting for the structural influence of neighboring local models.

Without message passing, the VAE-generated poisoned model updates may overlook subtle correlations and hidden patterns that naturally exist among genuine models in decentralized settings. As illustrated in Figs. 6, 7, 9, and 10, the malicious model updates appear more anomalous and less aligned with the benign ones, allowing the affected benign users in DFL to effectively counteract the attack by aggregating trustworthy models from their multiple benign

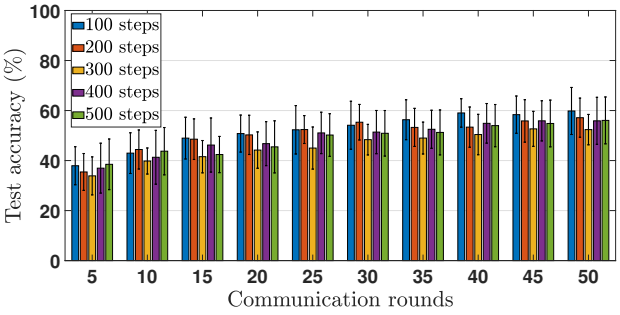


Fig. 12: Test accuracy of DFL under the proposed MPG-based UIP attack, when τ increases from 100 to 500.

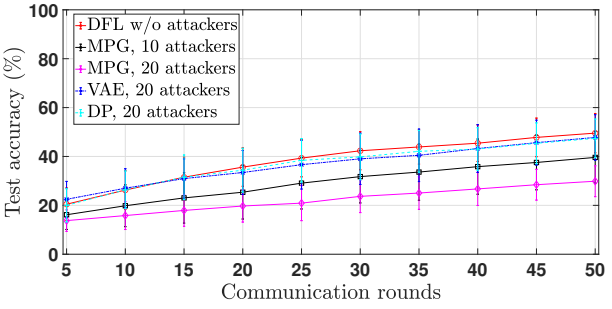


Fig. 13: The model test accuracy with the ring topology of DFL.

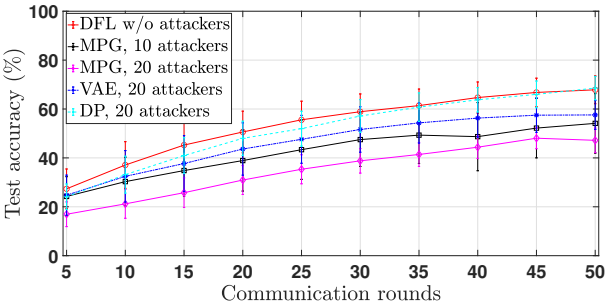


Fig. 14: The model test accuracy with the grid topology of DFL.

neighbors. This leads to relatively high test accuracy under the VAE attack. In contrast, the new MPG-based attack carefully crafts UIP model updates that maintain strong correlations with the benign models from the neighbors, while selectively isolating the targeted user's feature space. This targeted and context-aware manipulation leads to a more pronounced degradation in test accuracy, demonstrating superior attack effectiveness in DFL systems.

5) Impact of message passing steps τ: Fig. 12 shows an average test accuracy of DFL under the proposed MPG-based UIP attack, when the number of message passing steps τ increases from 100 to 500. We notice a non-linear behavior, where the lowest test accuracy of DFL occurs at 300 message-passing steps instead of continuously decreasing with increased τ. This can primarily be due to an over-smoothing effect in A-MPNN. Initially, increasing τ to up to 300 allows the malicious model to better aggregate and exploit fine-grained correlations among benign model updates, enhancing the effectiveness of the crafted poisoned model updates and lowering test accuracy.

When $\tau > 300$, the excessive message passing leads to diminishing returns, as the learned representations become overly smoothed and diluted, causing essential adversarial rewiring of $\mu^{\tau}_{n,n'}$ and subtle distinctions between malicious and benign features to weaken. As a result, the generated malicious updates lose their specificity and precision, reducing the UIP attack's effectiveness, which explains why the test accuracy no longer drops linearly and even begins to recover slightly after exceeding the 300 steps.

6) Scalability of The UIP Attack: To demonstrate the generalisability of our proposed attacking algorithm in larger DFL systems, we extend the number of benign users and malicious users to 100 and 20, respectively. As shown in Figs. 13 and 14, our MPG-based UIP attack remains consistently outperforming the baseline algorithms, DP-based and VAE-based attacks. Specifically, in the ring topology, the test accuracy under the proposed MPG-based attack is 17.9% and 18.4% lower than the VAE-based attack and the DP-based attack, respectively. In the grid topology, compared with VAE-based attack and DP-based attack, our attack achieves 10.4% and 20.4% lower than the two approaches, respectively. Moreover, the UIP attack's effectiveness increases with the number of malicious users. Adding more adversaries increases the fraction of UIP model updates circulating in the peer graph, which raises the probability that benign users receive and incorporate UIP features in the shared model updates. Adding more adversaries also enlarges the adversarial influence in the A-MPNN to learn and distort feature correlations, while permitting perturbations that reinforce each other across communication rounds.

7) Runtime Measurements: We measured the per-round runtime of the proposed MPG-based UIP attack on a large-scale DFL (20 attackers, 100 benign users). As shown in Fig. 15, under the grid topology, the attacker's average runtime per round lies roughly in the 1.4-1.9s range ($\approx$1.65s on average). As shown in Fig. 16, under the ring topology, the measured runtimes are slightly higher, about 1.5-1.9s per round ($\approx$1.70s on average). These results show that the dominant cost is the local A-MPNN message-passing and embedding updates (which scale with the number of local neighbors, embedding dimension κ, τ message-passing steps, and $\mathcal{B}$ layers); in our configuration the per-attacker overhead is therefore on the order of one to two seconds per communication round. Because this computation is performed locally by the attacker (no extra network-wide coordination required), it is parallelizable and can be reduced by tuning κ, τ or $\mathcal{B}$, or by batching message computations. Therefore, the MPG-based UIP attack is computationally feasible at the edge in our experiments (about 1.6-1.7s/round per attacker).

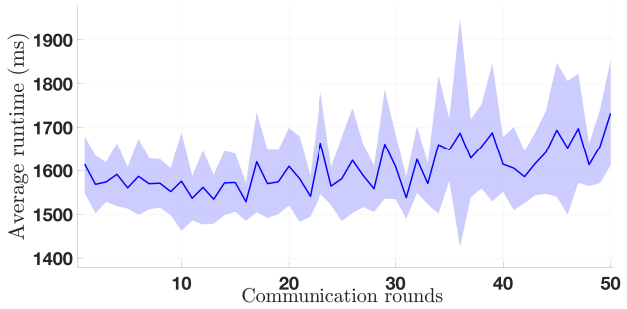


Fig. 15: Average runtime of the MPG-based UIP attack under the grid topology

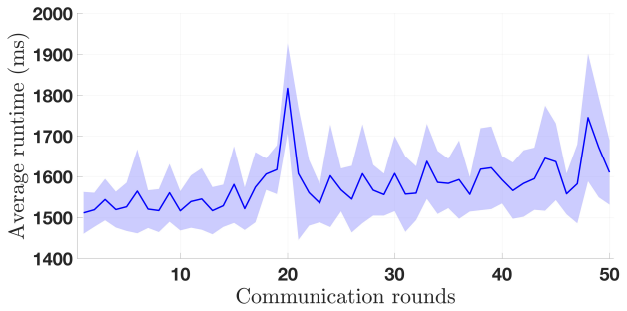


Fig. 16: Average runtime of the MPG-based UIP attack under the ring topology.

VI. Conclusions and Future Research

This paper proposed the MPG-based UIP attack, which strategically suppresses the influence of benign users' model updates in DFL to steer the shared model toward adversarial objectives. To address the complexity of the non-convex adversarial training problem, a graph signal processing framework was proposed to iteratively refine the malicious updates through alternating execution of the UIP mechanism and sub-gradient descent. Moreover, A-MPNN was jointly trained with sub-gradient descent to learn the structural dependencies among benign model updates. By adversarially reconstructing these relationships, A-MPNN maximizes reconstruction loss while keeping the compromised model updates undetectable. The attack was implemented in PyTorch and validated through experiments, demonstrating a significant reduction in the test accuracy and successful evasion of state-of-the-art defenses based on cosine similarity and Euclidean distance.

The proposed MPG-based UIP attack focuses on homogeneous DFL that is consistent with many typical application scenarios, such as collaborative learning among mobile devices of the same manufacturer, IoT sensors deployed in smart cities, or wearable sensors on patients in smart hospitals, where users share similar computational capabilities and model architectures. Within this adopted setting, our proposed UIP attack introduces a fundamentally novel threat model against DFL, exposing a critical vulnerability of DFL systems that had not been studied before. DFL deployments may also involve heterogeneous devices and non-IID data distributions, requiring adversaries to align updates across different feature spaces or adapt message passing to non-uniform topologies. Hence, extending the MPG-based UIP attack to the heterogeneous DFL presents a meaningful direction for future research.

In addition, developing effective defense mechanisms against the proposed MPG-based UIP attack is essential to enhance the robustness of DFL. Potential defenses should focus on detecting subtle and coordinated adversarial behaviors that manipulate structural relationships among model updates while maintaining statistical similarity to benign models. This calls for a deeper understanding of both the topological and dynamic properties of decentralized model interactions. Future work may explore adaptive, context-aware strategies that go beyond traditional similarity-based checks, aiming to preserve the integrity and diversity of benign contributions without compromising the decentralized nature of the system.

Acknowledgements

This work was supported by the CISTER Research Unit (UIDP/UIDB/04234/2020) and project ADANET (PTDC/EEICOM/3362/2021), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology); and also supported in part by the AXA Research Fund (AXA Chair for Internet of Everything at Koç University).

The authors would like to thank Dr. Petar Veličković (a Staff Research Scientist at Google DeepMind and Affiliated Lecturer at the University of Cambridge, https://petarv.com/) for his assistance with the formulation of the A-MPNN architecture, and constructive comments on the article.

References

[1] E. T. M. 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 & Tutorials*, 2023.

[2] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, 2024.

[3] G. Zhao, Y. Shen, C. Zhang, Z. Shen, Y. Zhou, and H. Wen, "Rgbe-gaze: A large-scale event-based multimodal dataset for high frequency remote gaze tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] G. Zhao, Y. Yang, J. Liu, N. Chen, Y. Shen, H. Wen, and G. Lan, "Ev-eye: Rethinking high-frequency eye tracking through the lenses of event cameras," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] K. Li, Y. Cui, W. Li, T. Lv, X. Yuan, S. Li, W. Ni, M. Simsek, and F. Dressler, "When internet of things meets metaverse: Convergence of physical and cyber worlds," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4148–4173, 2022.

[6] K. Li, B. P. L. Lau, X. Yuan, W. Ni, M. Guizani, and C. Yuen, "Towards ubiquitous semantic metaverse: Challenges, approaches, and opportunities," *IEEE Internet of Things Journal*, 2023.

[7] M. Fang, Z. Zhang, Hairi, P. Khanduri, J. Liu, S. Lu, Y. Liu, and N. Gong, "Byzantine-robust decentralized federated learning," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2874–2888.

[8] S. Tan, F. Hao, T. Gu, L. Li, and M. Liu, "Collusive model poisoning attack in decentralized federated learning," *IEEE Transactions on Industrial Informatics*, 2023.

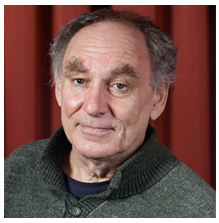
[9] L. Zhang, S. Qin, G. Feng, and Y. Peng, "Decentralized federated learning under free-riders: Credibility analysis," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2024, pp. 01–06.
[10] C. Bodnar, F. Frasca, Y. Wang, N. Otter, G. F. Montufar, P. Lio, and M. Bronstein, "Weisfeiler and lehman go topological: Message passing simplicial networks," in *Proceedings of International Conference on Machine Learning*. PMLR, 2021, pp. 1026–1037.
[11] Y. Qu, C. Xu, L. Gao, Y. Xiang, and S. Yu, "Fl-sec: Privacy-preserving decentralized federated learning using signsgd for the internet of artificially intelligent things," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 85–90, 2022.
[12] A. Gholami, N. Torkzaban, and J. S. Baras, "Trusted decentralized federated learning," in *Proceedings of IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 1–6.
[13] M. Cambus, D. Melnyk, T. Milentijevic, and S. Schmid, "Coordinate-wise median in byzantine federated learning," in *Proceedings of the International Workshop on Secure and Efficient Federated Learning*, 2025, pp. 1–6.
[14] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.
[15] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.
[16] P. Sun, X. Liu, Z. Wang, and B. Liu, "Byzantine-robust decentralized federated learning via dual-domain clustering and trust bootstrapping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24 756–24 765.
[17] V. Shejwalkar, A. Houmansadr, P. Kairouz, and D. Ramage, "Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1354–1371.
[18] X. Cao and N. Z. Gong, "Mpaf: Model poisoning attacks to federated learning based on fake clients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3396–3404.
[19] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *Proceedings of the 29th USENIX security symposium (USENIX Security)*, 2020, pp. 1605–1622.
[20] K. Li, J. Zheng, X. Yuan, W. Ni, O. B. Akan, and H. V. Poor, "Data-agnostic model poisoning against federated learning: A graph autoencoder approach," *IEEE Transactions on Information Forensics and Security*, 2024.
[21] K. Li, X. Yuan, J. Zheng, W. Ni, F. Dressler, and A. Jamalipour, "Leverage variational graph representation for model poisoning on federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
[22] J. Zheng, K. Li, N. Mhaisen, W. Ni, E. Tovar, and M. Guizani, "Exploring deep-reinforcement-learning-assisted federated learning for online resource allocation in privacy-preserving edgeiot," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 099–21 110, 2022.
[23] Y. Shi, L. Shen, K. Wei, Y. Sun, B. Yuan, X. Wang, and D. Tao, "Improving the model consistency of decentralized federated learning," in *Proceedings of International Conference on Machine Learning (ICML)*. PMLR, 2023, pp. 31 269–31 291.
[24] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
[25] W. Li, T. Lv, W. Ni, J. Zhao, E. Hossain, and H. V. Poor, "Decentralized federated learning over imperfect communication channels," *IEEE Transactions on Communications*, 2024.
[26] K. Li, X. Yuan, J. Zheng, W. Ni, and M. Guizani, "Exploring adversarial graph autoencoders to manipulate federated learning in the internet of things," in *IEEE International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2023, pp. 898–903.
[27] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1639–1654, 2022.
[28] Y. Miao, Z. Liu, H. Li, K.-K. R. Choo, and R. H. Deng, "Privacy-preserving byzantine-robust federated learning via blockchain systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2848–2861, 2022.
[29] K. Wei, J. Li, M. Ding, C. Ma, Y.-S. Jeon, and H. V. Poor, "Covert model poisoning against federated learning: Algorithm design and optimization," *IEEE Transactions on Dependable and Secure Computing*, 2023.
[30] K. Wang, Q. He, F. Chen, H. Jin, and Y. Yang, "Fededge: Accelerating edge-assisted federated learning," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 2895–2904.
[31] C. Cangea, B. Day, A. R. Jamasb, and P. Lio, "Message passing neural processes," in *ICLR Workshop on Geometrical and Topological Representation Learning*, 2022.
[32] L. Telyatnikov, M. S. Bucarelli, G. Bernardez, O. Zaghen, S. Scardapane, and P. Lio, "Hypergraph neural networks through the lens of message passing: a common perspective to homophily and architecture design," *arXiv preprint arXiv:2310.07684*, 2023.
[33] F. Di Giovanni, L. Giusti, F. Barbero, G. Luise, P. Lio, and M. M. Bronstein, "On over-squashing in message passing neural networks: The impact of width, depth, and topology," in *Proceedings of International Conference on Machine Learning*. PMLR, 2023, pp. 7865–7885.
[34] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "Mgae: Marginalized graph autoencoder for graph clustering," in *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, 2017, pp. 889–898.
[35] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, "Topology-aware federated learning in edge computing: A comprehensive survey," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–41, 2024.
[36] M. Ma, T. Li, and X. Peng, "Beyond the federation: Topology-aware federated learning for generalization to unseen clients," *arXiv preprint arXiv:2407.04949*, 2024.
[37] M. T. Hossain, S. Badsha, H. La, S. Islam, and I. Khalil, "Exploiting gaussian noise variance for dynamic differential poisoning in federated learning," *IEEE Transactions on Artificial Intelligence*, 2025.

Kai Li (S'09–M'14–SM'20) received the B.E. degree from Shandong University, China, in 2009, the M.S. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2010, and the Ph.D. degree in computer science from the University of New South Wales, Sydney, Australia, in 2014. He is currently an Assistant Professor in the Department of Information Technology, College of Computing and Software Engineering, Kennesaw State University, Marietta, Georgia, USA. Funded by the CMU-Portugal Visiting Faculty and Researchers Program, Dr. Li was a Visiting Scholar in the Department of Electrical and Computer Engineering, College of Engineering, Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania, from November to December 2025. From 2024 to 2025, he was a Visiting Research Scholar with the School of Electrical Engineering and Computer Science, TU Berlin, Germany. From 2016 to 2025, he served as a Senior Research Scientist at the CISTER Research Centre, Porto, Portugal, and concurrently as a CMU-Portugal Research Fellow, jointly supported by Carnegie Mellon University and the Foundation for Science and Technology (FCT), Lisbon, Portugal. From 2023 to 2024, he was a Visiting Research Scientist with the Division of Electrical Engineering, Department of Engineering, University of Cambridge, UK. In 2022, he was a Visiting Research Scholar with the CyLab Security and Privacy Institute at CMU. Prior to these, he worked as a Post-Doctoral Research Fellow with the SUTD-MIT International Design Centre, Singapore University of Technology and Design (SUTD), Singapore, from 2014 to 2016. He also held positions as a Visiting Research Assistant with the ICT Centre, CSIRO, Brisbane, Australia, from 2012 to 2013, and as a Research Assistant with the Mobile Technologies Centre, The Chinese University of Hong Kong, from 2010 to 2011. He has served as an Associate Editor for several journals, including *Internet of Things* (Elsevier) since 2024, *Nature Computer Science* (Springer) since 2023, *Computer Communications* (Elsevier) and *Ad Hoc Networks* (Elsevier) since 2021, and IEEE ACCESS from 2018 to 2024.

Pietro Liò received the M.A. degree from the University of Cambridge, the Ph.D. degree in complex systems and non-linear dynamics from the Department of Engineering, School of Informatics, University of Firenze, Italy, and the Ph.D. degree in (theoretical) genetics from the University of Pavia, Italy. He is currently a Full Professor of computational biology with the Computer Laboratory, University of Cambridge, and a member of the Artificial Intelligence Group. His research interests include bioinformatics, computational biology modeling, and machine learning to integrate various types of data (molecular and clinical, drugs, social, and lifestyle) across different spatial and temporal scales of biological complexity to address personalized and precision medicine.

Wei Ni (M'09–SM'15–F'24) received the B.E. and Ph.D. degrees in Electronic Engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is a Senior Principal Research Scientist at CSIRO and a Conjoint Professor at the University of New South Wales, Sydney, Australia. He was a Deputy Project Manager at the Bell Labs, Alcatel/Alcatel-Lucent from 2005 to 2008, and a Senior Researcher at Devices R&D, Nokia from 2008 to 2009. He has coauthored three books, eleven book chapters, over 550 technical papers, 27 patents, and ten standard proposals accepted by IEEE. His research interests include machine learning, online learning, stochastic optimization, and their applications to system efficiency and integrity. Dr. Ni has been an Editor for IEEE Transactions on Wireless Communications since 2018, IEEE Transactions on Vehicular Technology since 2022, IEEE Transactions on Information Forensics and Security and IEEE Communication Surveys and Tutorials since 2024, and IEEE Transactions on Network Science and Engineering since 2025. He served first as Secretary, then Vice-Chair and Chair of the IEEE VTS NSW Chapter from 2015 to 2022, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTC-Spring 2016, Publication Chair for BodyNet 2015, and Student Travel Grant Chair for WPMC 2014.

Falko Dressler (F'17) is a full professor and Chair for Telecommunication Networks at the School of Electrical Engineering and Computer Science, TU Berlin. He received his M.Sc. and Ph.D. degrees from the Dept. of Computer Science, University of Erlangen in 1998 and 2003, respectively. Dr. Dressler has been associate editor-in-chief for IEEE Trans. on Network Science and Engineering, IEEE Trans. on Mobile Computing and Elsevier Computer Communications as well as an editor for journals such as IEEE/ACM Trans. on Networking, Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. Dr. Dressler is an IEEE Fellow, an ACM Fellow, and an AAIA Fellow. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include next generation wireless communication systems in combination with distributed machine learning and edge computing for improved resiliency. Application domains include the internet of things, cyber-physical systems, and the internet of bio-nano-things.



Jon Crowcroft (F'16) received the degree in physics from Trinity College, University of Cambridge, Cambridge, U.K., in 1979, and the M.Sc. and Ph.D. degrees in computing from University College London, London, U.K., in 1981 and 1993, respectively. From 2016 to 2018, he was the Programme Chair with the Alan Turing Institute, U.K. National Data Science and AI Institute, London, U.K. He is currently a Researcher with the Alan Turing Institute. Since October 2001, he has been a Marconi Professor of Communications Systems with the Department of Computer Science and Technology, University of Cambridge. His research interests include internet support for multimedia communications, scalable multicast routing, practical approaches to traffic management, the design of deployable end-to-end protocols, opportunistic communications, social networks, privacy-preserving analytics, and techniques and algorithms to scale infrastructure-free mobile systems. He is a fellow of the Royal Society, ACM, British Computer Society, IET, and the Royal Academy of Engineering.

Ozgur B. Akan (F'16) received the PhD degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2004. He is currently the head of Internet of Everything (IoE) Group, with the Department of Engineering, University of Cambridge, U.K., and the director of Centre for Next-Generation Communications (CXC), KoÇuniversity, Turkey. His research interests include wireless, nano, and molecular communications, and Internet of Everything.