Backdoor Attacks in Peer-to-Peer Federated Learning

Gökberk Yar, Simona Boboila, Cristina Nita-Rotaru, Alina Oprea
Northeastern University, Khoury College of Computer Sciences

***Abstract*—Most machine learning applications rely on centralized learning processes, opening up the risk of exposure of their training datasets. While federated learning (FL) mitigates to some extent these privacy risks, it relies on a trusted aggregation server for training a shared global model. Recently, new distributed learning architectures based on Peer-to-Peer Federated Learning (P2PFL) offer advantages in terms of both privacy and reliability. Still, their resilience to poisoning attacks during training has not been investigated. In this paper, we propose new backdoor attacks for P2PFL that leverage structural graph properties to select the malicious nodes, and achieve high attack success, while remaining stealthy. We evaluate our attacks under various realistic conditions, including multiple graph topologies, limited adversarial visibility of the network, and clients with non-IID data. Finally, we show the limitations of existing defenses adapted from FL and design a new defense that successfully mitigates the backdoor attacks, without an impact on model accuracy.**

I. Introduction

Recently, machine learning (ML) has transformed a variety of real-life applications including self-driving cars [1], recommendation engines [2], and personalized health [3]. A common thread to all these applications is data centralization, where a significant amount of data is collected for training ML models, a process that introduces risks to the privacy of users contributing their datasets. Several regulations such as the General Data Protection Regulation (GDPR) [4] and the California Consumer Privacy Act (CCPA) [5] were specifically designed to protect data privacy.

To address the privacy concerns of centralized learning, McMahan et al. [6], and Konecny et al. [7] proposed Federated Learning (FL), a paradigm that trains ML models in a distributed fashion. In FL, personal data never leaves the device. Instead, devices individually update a global model and share their model updates with a central server, which aggregates them and re-distributes the new global model to the clients. Privacy in FL can be enhanced with Multi-Party Computation (MPC) [8] and differential privacy [9], but most FL deployments do not utilize these technologies and are vulnerable to privacy attacks [10], [11].

In response to the privacy and reliability risks of FL with a single aggregation server, protocols for Peer-to-Peer Federated Learning (P2PFL) have been proposed [12]–[18]. In P2PFL nodes communicate with their peers in the network and aggregate the model updates received from their peers. Multiple aspects of P2PFL have been studied, including adversarial settings (Byzantine [19] vs non-Byzantine [20]), the P2P network topology (complete graphs [19] vs non-complete graphs [14]), and the output of the learning protocol (global model learned by all peers [19] vs. personalized models [12]).

Real-life deployments of FL [21], [22] raised concerns about adversarial attacks such as poisoning. While such attacks have been extensively studied in centralized learning [23]–[31], they are more feasible in Federated Learning because adversaries can own or compromise mobile devices and participate in the FL training process. Attack vectors in FL include data poisoning [32], and model poisoning [33], [34], where the attacker aims to pursue availability, targeted, or backdoor attacks. The objective of an availability attack is to decrease model accuracy and its utility [35], [36], while targeted and backdoor attacks [37], [38] impact only a subpopulation of samples without dropping the overall model accuracy.

To date, sophisticated data poisoning attacks with stealthy objectives such as backdoor and targeted attacks have not been studied in P2PFL. In addition, most P2PFL systems [15], [16], [19] either consider that the P2P network topology is a complete graph or do not mention the topology at all. Complete network topologies have inherent scalability issues and extremely high network bandwidth costs. In practical deployments of P2PFL, such connectivity assumptions are almost infeasible to satisfy, thus it is important to study attacks in P2PFL considering realistic network topologies.

In this paper, we propose and evaluate backdoor attacks in P2PFL where the adversary controlling a small set of nodes, has two goals: remaining stealthy, and achieving high attack success on samples with the backdoor pattern. These goals are conflicting and difficult to achieve simultaneously: as the attack becomes more successful it likely causes degradation in model accuracy, which can be detected by defenders. We consider realistic graph topologies [39]–[42] in which attackers can leverage information about the graph structure to increase the effectiveness of their attack. Specifically, our contributions are:

- We present a modular architecture for P2PFL that supports diverse network topologies and separates the learning graph from the communication graph to study poisoning attacks in realistic settings. Our current implementation uses GossipSub as the communication layer and P2P gradient averaging as the learning protocol. All code is publicly released[1].
- We propose backdoor attacks in P2PFL and introduce new attack placement strategies based on graph centrality metrics.

[1]https://github.com/gokberkyar/BP2PFL

We show that a small number of attackers (5%) placed in the graph strategically, is sufficient to perform a backdoor attack with high attack success without decreasing the model accuracy for multiple graph topologies. We show that backdoor attacks can further be amplified by the attacker causing network failures that result in missed peer updates. We also demonstrate that an attacker with partial visibility into the network (e.g., 20% of the nodes) can still successfully introduce a backdoor in the model.

• Our paper is the first extensive study on the impact of P2PFL backdoor attacks on various learning network topologies. Our study shows that the Barabasi-Albert scale-free network is the most vulnerable due to the presence of "hubs" (highly connected nodes). In a strategically placed attack, compromising the hub nodes will result in a particularly strong backdoor attack.

• We introduce a new P2P defense based on weighting a node's contribution higher than the contributions of its peers when training each model. We propose a defense that uses two different clipping norms, one for peer updates, and one for local models, and thus bounds the contribution of a node's neighbors to effectively prevent backdoor attacks.

• We analyze the impact of label imbalance in non-IID settings, where peers have different data distributions. We show that the non-IID model converges slower than IID to correctly classify clean data due to heterogeneity in local updates, however, the attack is still able to induce a high level of misclassification.

II. Threat Model

Adversarial goal. An attacker may perform poisoning attacks with different goals, such as availability, targeted, or backdoor attacks. In this paper, we focus on backdoor attacks due to their particularly insidious nature and the feasibility of having malicious participants in P2P protocols. In backdoor attacks, the adversary has two goals: 1) Remaining stealthy, and 2) Achieving high attack success on backdoored samples. These objectives are conflicting and difficult to achieve simultaneously: as the attack becomes more successful at inducing misclassifications on backdoored samples, it likely causes degradation in model accuracy, which can be detected by defenders.

Adversary participation in the P2PFL protocol. As P2PFL is an open system, we will assume the attacker either controls or compromises a number of k peers, where $k < N$, with N the total number of peers in the system. Compromising a peer in this context might be easier than compromising a well-protected aggregation server in FL.

Attacker capabilities. We assume that the attacker has full control over compromised peers. More precisely, the attacker can add, modify, or delete training data samples, modify and deviate from the machine learning algorithm, as long the final model update has the same vector dimension as the model updates sent by benign users. For example, the attacker can increase the number of local epochs used in training, change the learning rate of the model, or even apply a new learning objective function. The attacker can only observe model updates received from his peers.

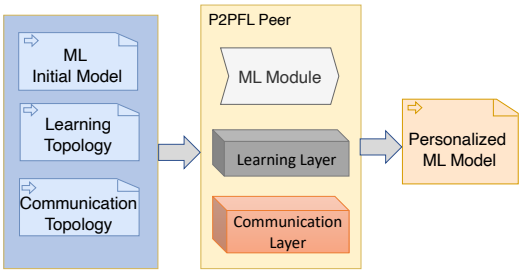


Fig. 1. P2PFL Architecture Overview: A peer has 3 roles: 1) Forwarding network packages (Communication Layer); 2) Sending and receiving ML updates to data peers (Learning Layer); 3) Running ML training on local dataset, aggregating updates received by the Learning Layer, and sharing back with the Learning Layer for the next round (ML Module).

Additionally, the attacker has network-level capabilities. In our strongest attack model, the attacker has full visibility of the peer connections and may use this knowledge to select and compromise the most critical nodes in the communication graph. Powerful attackers are relevant in evaluating and comparing mitigation strategies. We also consider a more relaxed adversarial model in which the adversary has partial visibility over the network, by observing a small fraction (e.g., 20%) of the nodes in the communication graph.

Attacker strategy. Backdoor attacks have three components in P2PFL: pre-training phase, training time phase, and inference phase. In the pre-training phase, the adversary injects the desired backdoor (computed via standard methods [43]) into a subset of the training data at compromised peers. Specifically, if the attacker compromises peer i, has full access to D_i and picks a Poisoning Data Ratio (PDR), then injects PDR $\cdot |D_i|$ backdoored samples into the training dataset. In the training phase, the adversary modifies the hyper-parameters of the local training and modifies model weights to conduct the model poisoning attack, introduced for FL [34]. Finally, at inference time, if the adversary wishes to change the prediction of a certain sample, it injects the desired backdoor pattern.

III. P2P Federated Architecture

We introduce the P2PFL architecture, the design decisions, and the Personalized Peer-to-Peer Averaging Algorithm.

A. P2PFL Architecture Design

As previous P2P learning systems assumed a connected graph for communication [19], we consider realistic network topologies and create a modular architecture shown in Figure 1 that separates the learning and communication graphs.

We define the *learning topology* as a graph in which peers are connected if they exchange model updates during training. We call these peers **learning peers**. We define the *communication topology* as the graph in which peers are connected if they can route network packets between them. The learning and communication topologies are shown in Figure 2. To distinguish the communication and learning topologies, we

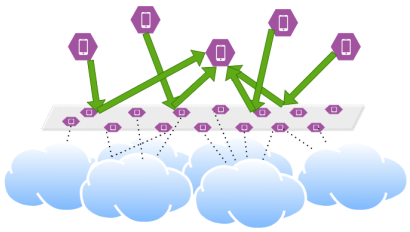


Fig. 2. Communication topology is shown by the gray plane, learning topology uses the underlying communication topology to exchange updates.

design P2PFL as an application that runs over the GossipSub protocol [44]. GossipSub constructs a mesh where nodes can publish their updates or subscribe to other nodes' updates, and has been shown to successfully balance excessive bandwidth consumption and fast message propagation [44]. When node A wants to communicate to node B, messages cannot be directly passed from A to B with a TCP connection, but rather A publishes its update into a channel that B subscribed into. We call these peers **communication peers**. This design choice provides better network bandwidth utilization and robustness to network failures.

B. P2P Gradient Averaging Algorithm

While our architecture supports different ML algorithms, we use P2P Gradient Averaging (Algorithm 1), based on the standard Federated Averaging algorithm [6], [45]. Each node trains a personalized model using the updates received from its neighbor set in the learning graph.

Algorithm 1: Personalized Peer-to-Peer Gradient Averaging

Data: Local Dataset D, rounds T_N, peer set S
Function `P2PAverage()`:
  $f_0 = \text{GETINITIALMODEL}(0)$
  for $t \in [1, T_N]$ **do**
    // Compute local update using SGD
    $A^t = \text{COMPUTELOCALUPDATE}(f_{t-1}, D)$
    // In parallel send and receive updates
    $U^t = \text{GETUPDATES}(S)$
    $\text{SENDUPDATE}(A^t, S)$
    $f_t = \text{AGGREGATE}(A^t, U^t)$
  return f_n

Specifically, each node participating in the protocol owns its private training dataset D, a set of peer nodes S, to which the node can exchange model updates and its personalized model f_t at round t. Each node synchronously trains its personalized model f_t at round t by computing its local update A^t using its private dataset D, and then aggregating with the previous round's personalized model f_{t-1}, and model updates received from its peers U^t. For the aggregation step, we average all the model updates received from the neighboring peers. The framework supports the use of more advanced Byzantine robust aggregation functions such as KRUM [46] and Bulyan [47] that have been proposed for poisoning defense in FL. We adapt the Trimmed Mean [48] and gradient clipping [34], [37] FL defenses to the P2PFL setting.

IV. Backdoor Attacks on P2P Federated Learning

In this section we describe the backdoor attacks we propose for P2PFL and discuss different attack strategies that select relevant peers based on graph centrality metrics.

A. Backdoor Attack on P2PFL

We assume the adversary has full access to the peer's private training data, model weights, and other training parameters in each compromised peer and runs a backdoor attack on each compromised peer using Algorithm 2. The attacker uses the BadNets attack by Gu et al. [43], amplified by a model poisoning attack [34], [37]. The attacker chooses a Poisoning Data Ratio and the Boosting Factor to amplify the contribution of the local model. Increasing the Poisoning Data Ratio and Boosting Factor increases the attack success, but also causes a drop in test accuracy as a side effect.

Algorithm 2: P2P Backdoor Attack Single Node

Data: Target Node $target$, Local Dataset D, rounds T_N, peer set S, poison data rate PDR, boosting factor B, target class C_t
Function `AttackSingle`$(target, PDR, B, C_t)$:
  $f_0 = \text{GETINITIALMODEL}(0)$
  $D^* = \text{BACKDOORDATASET}(D, PDR, C_t)$
  for $t \in [1, T_N]$ **do**
    // Compute local update on Backdoored Dataset
    $A^t = \text{COMPUTELOCALUPDATE}(f_{t-1}, D^*)$
    // Do model poisoning attack by boosting
    $A^{t^*} = \text{MODELPOISONUPDATE}(A^t, B)$
    // In parallel send and receive updates
    $U^t = \text{GETUPDATES}(S)$
    // Send malicious updates
    $\text{SENDUPDATE}(A^{t^*}, S)$
    $f_t = \text{AGGREGATE}(A^{t^*}, U^t)$

B. Stronger Structural Graph Attacks

The P2PFL protocol runs on a non-complete graph topology, thus not all peers have an equal impact when used by the attacker. One natural question that the adversary must answer is how to choose the k peers for attack, among all n peers available in the system. For example, if the attacker has budget for compromising only 5% of all nodes, how can the adversary maximize its adversarial goal by carefully selecting those attacker nodes?

One simple strategy for the attacker is to select the adversarial nodes randomly. However, considering the P2P nature of the system, a natural approach is to select nodes that are well connected in the graph and have high centrality measures. We introduce several attack strategies based on four well-known graph centrality metrics: maximum degree, Effective Network Size [49], PageRank scores [50], and maximum clustering coefficient [51].

Maximum degree. Nodes with the highest degree in the graph allow the attacker to propagate malicious updates to a large number of neighbors.

ENS score. The effective network size (ENS) of a node's ego network can be computed as:

$$e(u) = n - 2t/n \tag{1}$$

where t is the number of ties in the ego network (not including ties to the node itself) and n is the number of nodes in the ego network. A recent work on cyber network resilience against self-propagating malware [52] observed that nodes with the highest ENS tend to act as bridges between two dense clusters and monitoring them prevents attack spreading. Our insight is that instead of using ENS for robustness, we select nodes with largest ENS scores to enable the attacker to traverse bridges in the network and compromise different clusters.

PageRank score. The PageRank score computes a ranking of the nodes in graph G based on the structure of the incoming links [50] and provides a metric of centrality and node importance in the graph. As PageRank showed empirical success on sparse graphs and the attacker's goal is to identify critical nodes, we leverage nodes with highest PageRank score as a viable attack selection strategy.

Maximum clustering coefficient. For unweighted graphs, the clustering coefficient of a node is the ratio of possible triangles through that node:

$$c_u = \frac{2T(u)}{\mathsf{deg}(u)(\mathsf{deg}(u) - 1)} \tag{2}$$

where $T(u)$ is the number of triangles through node u and $\mathsf{deg}(u)$ is the degree of node u [51]. We select this metric as nodes with highest clustering coefficient tend to have a more connected local neighborhood where the malicious updates can propagate.

C. Quantifying Attack Success

The attacker has two objectives: 1) Performing a successful backdoor attack, and 2) Remaining stealthy so that the attack is not detected by monitoring the model's accuracy. We quantify these goals with two metrics used in the poisoning literature: (1) **Attack Success**, denoting the fraction of poisoned samples that were incorrectly classified as belonging to attacker's target class, and (2) **Test Accuracy**, representing the fraction of clean samples that were correctly classified. These two metrics are averaged across all participants. To evaluate the attack we use an auxiliary test set partitioned into two non-overlapping subsets: a clean dataset, and a backdoored dataset, on which we compute the test accuracy and attack success, respectively.

V. Experimental Evaluation

In this section we evaluate the effectiveness of the attacks presented in Section IV. We first introduce our experimental setup, then evaluate the effectiveness of attacks under several settings, by seeking to answer the following questions:

- What node selection strategy provides most benefit to the adversary?
- What graph topology is more impacted by attacks?
- How does compromising more peers affect the attack?
- What is the impact of link failures on backdoor attacks?
- How does the data distribution impact the attack?
- What is the effect of constraining the adversary to a limited view of the network?

A. Experimental Setup

Network topology. We study three representative network topologies, which have been widely used to model complex networks: (i) Random graphs (Erdos-Renyi) [53]; (ii) Small-world graphs (Watts-Strogatz) [54]; and (iii) Scale-free graphs (Barabasi-Albert) [55].

Small-world graphs are characterized by a small average path length and high clustering coefficient, properties that have been observed in real-world networks [56]. In both random and small-world networks, nodes have comparable degrees, and, thus, the average can be viewed as the "scale" of the network. In contrast, in scale-free networks, the fluctuations from the average are large, with a few highly connected nodes serving as "hubs", while the vast majority have low degrees. The Internet is an example of a scale-free network, where the degree distribution is shaped by the "preferential attachment" to a small number of popular hubs [55].

Table I summarizes the parameters of the three topologies used in this study for a 60-node size network. We also experimented with smaller and larger networks from 30 to 100 nodes.

Parameter	Erdos R.	Watts S.	Barabasi A.	Complete
# nodes	60.00	60.00	60.00	60.00
# edges	166.00	360.00	576.00	1770.00
Mean Degree	5.53	12.00	19.20	59.00
Density	0.09	0.20	0.33	1.00
Diameter	5.00	3.00	3.00	1.00
Radius	3.00	2.00	2.00	1.00
Mean Distance	2.56	1.88	1.68	1.00
Transivity	0.10	0.24	0.39	1.00
Clustering coef.	0.09	0.24	0.41	1.00

TABLE I
CHARACTERISTICS OF NETWORK TOPOLOGIES USED IN THIS STUDY.

Datasets. We used the EMNIST [57], FashionMNIST [58] and MNIST [59] datasets featuring 28×28 pixel images labeled to one of 10 classes. The partitioning method among peers has a large impact on the ML model, generating peer datasets that fall into two broad categories: independently and identically distributed (IID), and non-independent and identically distributed (non-IID). Non-IID data distribution is a common challenge in FL [60]. In our paper, we analyze the attack performance in both IID and non-IID settings. In the IID setting, each client receives an equal number of samples of each label, while in the non-IID setting, each client is allocated a proportion of the samples of each label according to the Dirichlet distribution [61].

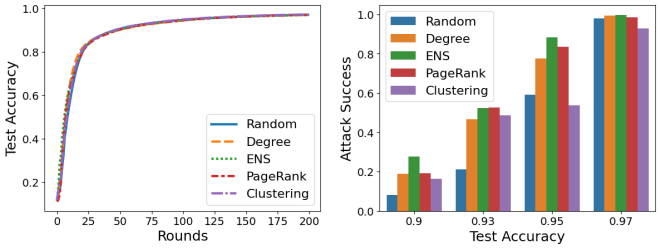
Parameters. The attack is characterized by the following parameters: k, the number of malicious peers; PDR, the ratio of

poisoned data over total samples in a malicious peer; boosting factor used in the model poisoning attack; adversarial training epoch count; and target class. We experimented with multiple values of these parameters. Here, we discuss the attack impact for various numbers of adversaries (k), while fixing the PDR to 0.5, the boosting factor to 10, adversarial training epoch count to 5, and the target class to 2. Results shown for various strategies and parameters are averaged over three different runs.

Default configuration. Unless otherwise specified, the experiments use the PageRank selection strategy on Watts Strogatz 60-node topology with 5% adversarial nodes. Furthermore, our default configuration uses the EMNIST dataset. Each peer receives a total of 5200 samples during training, with an equal number of samples per class (IID distribution).

B. Adversary's Node Selection Strategy

We evaluate the success of the attack for the node selection strategies introduced in Section IV-B: Random, Degree, ENS, PageRank, and Clustering. Figure 3 shows the evolution of P2PFL model's performance across training rounds. The test accuracy of the backdoored model on clean data converges similarly for all five attack strategies, and eventually reaches 0.97. In practice, the training may stop after a certain accuracy has been reached. Therefore, in Figure 3b we evaluate the attack success at high test accuracies. We notice a significant difference between the four attack strategies for accuracies in the low 90s (i.e., 0.9, 0.93), with the centrality-based methods Degree, ENS and PageRank being consistently on top, generally twice more successful than Random at misclassifiying the adversarial test samples. Once the model converges to 0.97 accuracy, the attack is highly successful regardless of the node selection method.

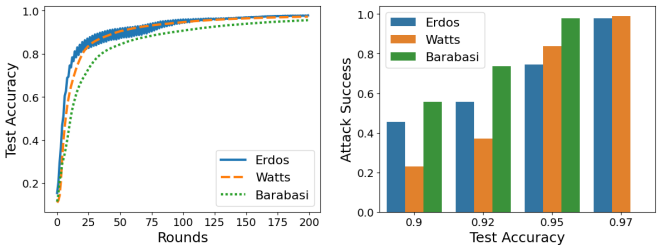


(a) Test Accuracy (b) Attack Success

Fig. 3. Adversary's node selection strategy: (a) accuracy of the backdoored model on clean test data. (b) attack success (y-axis) of various strategies after the backdoored model has reached high accuracy (x-axis).

C. Network Topology

We study the impact of the network topology on the attack. We compare the three types of graphs described in Table I: Erdos Renyi, Watts Strogatz, and Barabasi Albert, under a PageRank-based attack strategy. These graph models are typically used to analyze the behavior of social media and cellular network graphs, with previous work suggesting that P2PFL networks will most likely be small-world networks [39]–[42]. Figure 4a presents the accuracy of the backdoored P2PFL model across multiple rounds of training, and Figure 4b illustrates the attack success after the model has reached high accuracy (≥ 0.9) on clean data. These results point out a major insight: Barabasi scale-free network is the most vulnerable, due to the presence of highly connected nodes (hubs) that are selected with PageRank as the target of attack. The backdoored model becomes highly successful within 200 rounds of training on all topologies, learning to classify clean samples correctly (accuracy ≥ 0.95), but also to misclassify poisoned samples (0.99 attack success).

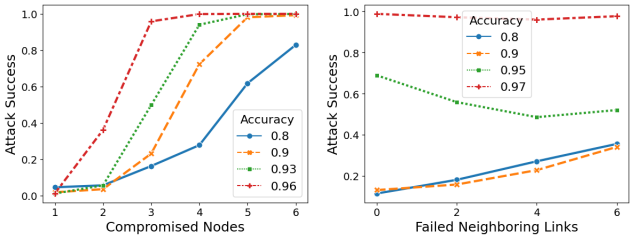


(a) Test Accuracy (b) Attack Success

Fig. 4. Impact of network topology on attack performance: (a) accuracy evolution; (b) attack success (y-axis) on various topologies after the backdoored model has reached high accuracy (x-axis). Within 200 rounds of training, the Barabasi topology has not exceeded 0.95 test accuracy (hence, the figure omits Barabasi at 0.97 accuracy).

D. Scaling Up the Attack

Figure 5a shows that the attack scales well with the number of adversarial nodes. Given a desired test accuracy, (i.e., 0.8, 0.9, 0.93, 0.96 in the figure), we evaluate the attack success for 1 to 6 compromised nodes selected with the PageRank strategy, on the 60-node Watts Strogatz topology. Note that our previous experiments have used only 3 adversarial nodes (i.e., 5%) while still delivering high performance. We also analyzed scaling to larger networks of 80 and 100 nodes, with expected results (not shown in figure): the convergence speed slightly decreases as the system scales up, however the attack is still highly successful.



(a) Adversarial Nodes (b) Fault Tolerance

Fig. 5. (a) Increasing the number of compromised nodes within a 60-node network. (b) Increasing the number of failed connections to neighbors (i.e., missed updates), with 3 compromised nodes.

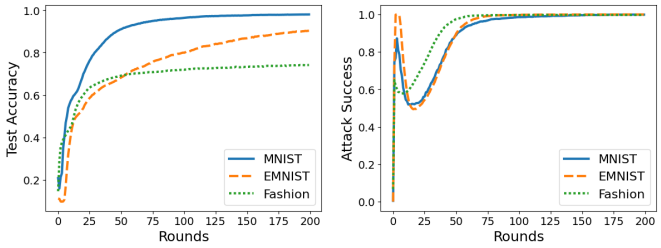
E. Fault Tolerance

As P2PFL is a collaborative distributed system, node failures are inevitable, and any P2PFL system should be tolerant to failures. In our study, we assume failures affect random nodes,

and result in missed peer updates that do not contribute to the learned model. Figure 5b measures the impact of 0, 2, 4, and 6 failed neighboring links per peer. Missing updates is generally in attacker's favor at lower accuracies of 0.8 and 0.9. As the model learns to classify clean data better (accuracy of 0.95 and 0.97), missed updates do not continue to aid the attack.

F. Impact of Dataset

In the previous experiments, we have shown that the P2P backdoor attack is highly successful on the EMNIST dataset. In this section, we explore its transferability to other datasets: MNIST and Fashion MNIST. These latter datasets are about $5\times$ smaller than EMNIST, therefore we reduce the number of peers to 30. We ensure that training is carried out on the same number of samples (based on the size of the smallest training dataset, MNIST), i.e., 1800 samples per client.



(a) Test Accuracy (b) Attack Success

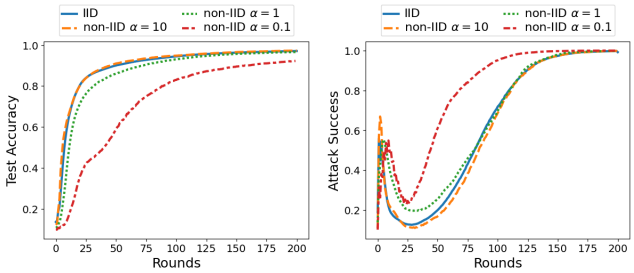
Fig. 6. Datasets comparison: (a) accuracy and (b) attack success of the backdoored P2PFL model. Configuration: 30 peers, 3 malicious nodes, Watts-Strogatz topology with PageRank attack.

Figure 6a presents the evolution of accuracy while training the P2PFL backdoored model. We observe that the model converges at different speeds for the various datasets. After 200 rounds of training, Fashion MNIST reaches 0.76 accuracy, EMNIST 0.91 and MNIST 0.98. We further note that the PageRank-based attack is highly successful at misclassifying poisoned samples on all datasets (Figure 6b), demonstrating that backdoor attacks are a valid threat for multiple applications. We notice an early peak in attack success when accuracy is still very low, indicating that malicious nodes (whose impact is amplified by gradient boosting) are more effective in the first rounds, before their contribution is offset by the honest majority. Reducing the boosting factor (we experimented with values of 10, 5, and 1) or the number of malicious peers has the effect of decreasing this early peek.

G. Impact of Data Distribution

We next compare the attack's performance (PageRank strategy) in IID settings (where nodes have an equal number of samples of each label), against non-IID settings (where nodes are allocated different numbers of samples of each class). We model the label imbalance of non-IID using the Dirichlet distribution [61], which is a common choice for simulating real-world data partitioning [60], [62]. The Dirichlet distribution is denoted $Dir(\alpha)$. The concentration parameter α controls the degree of similarity between peers. As $\alpha \to \infty$, peers' distributions become identical, whereas as $\alpha \to 0$, distributions become extremely imbalanced, with each class residing on separate peers.

Figure 7 compares IID against non-IID for three values of α: 10, 1, and 0.1. We observe that learning becomes more challenging in non-IID settings as the class imbalance increases (i.e., for smaller α). Heterogeneity in the peers' local datasets leads to large variations in local updates performed by peers [60]. As a results, accuracy convergences slower (Figure 7a) and the attack is more successful in inducing misclassifications (Figure 7b). In non-IID settings, where the class distribution is more skewed towards a subset of peers, we end up with fewer honest peers holding enough correctly labeled samples of each class. Thus, the honest updates are overpowered by the boosted adversarial updates, and a correct prediction for samples belonging to the target class is more difficult to learn (see non-IID $\alpha = 0.1$ from Figure 7b).



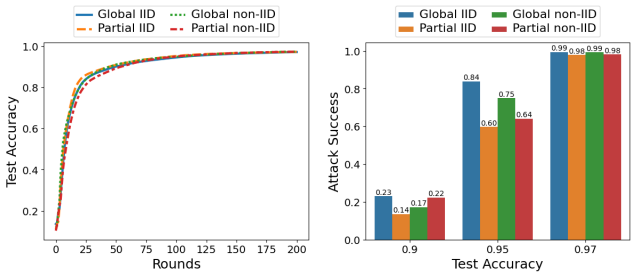
(a) Test Accuracy (b) Attack Success

Fig. 7. Impact of data distribution: IID and non-IID settings for three values of the concentration parameter α: 10, 1, and 0.1.

H. Constrained Adversary with Partial View of the Graph

In this section, we evaluate a constrained adversarial strategy, in which the attacker's view is restricted to a subset of the network. In our analysis, the observable subgraph represents 20% of the nodes in the network. The initial visible node is randomly chosen, while other nodes are added to the observable subset based on an exponential decay formula, p^d, were p is a probability parameter and d is the depth (i.e., number of hops) from the initial node. In our experiments, nodes that are further than 3-hops away ($d > 3$) have a zero probability of joining the observable subset ($p = 0$), otherwise $p = 0.5$. The attacker is restricted to applying his node selection strategies to the observable subset.

Figure 8 compares the constrained attacker (Partial view) against an attacker with full view of the network (Global view) under the PageRank-based attack. We observe that accuracy on clean test data converges similarly, regardless of attacker's view (Figure 8a), and reaches 0.97 in all cases. Figure 8b analyzes the attack success of the backdoored P2PFL model at high accuracies (≥ 0.9). For IID, the Global view consistently achieves higher attack performance compared to the Partial view. However, non-IID settings (illustrated for concentration parameter $\alpha = 10$) present a high variability, with the data distribution having a stronger impact on attacker's success than the observability restriction. Once the model has converged,



(a) Test Accuracy (b) Attack Success

Fig. 8. Constraining the adversary's view. We compare the Global with the Partial View in IID and non-IID ($\alpha = 10$) settings. (a) test accuracy on clean data; (b) attack success after the backdoored model has reached high accuracy.

non-IID and IID achieve the same level of attack success: 0.99 with Global view and 0.98 with Partial view.

VI. Defense

We first discuss standard defenses against poisoning attacks in FL and show empirical evidence that these defenses are ineffective, achieve slower convergence, and reduce the model accuracy. We then propose a new defense against poisoning in P2PFL and show experimentally that our defense counteracts backdoor attacks without a significant drop in model accuracy.

A. Existing Defenses

Existing defenses in FL rely on either robust aggregation functions [46]–[48] that exclude some of the outlying client updates from the model, or gradient clipping techniques [19], [37], [63] that limit the contributions of each client to the model update. We select a defense from each class: Trimmed Mean [48] and gradient clipping [37], adapt them to P2PFL, and evaluate them against the backdoor attacks.

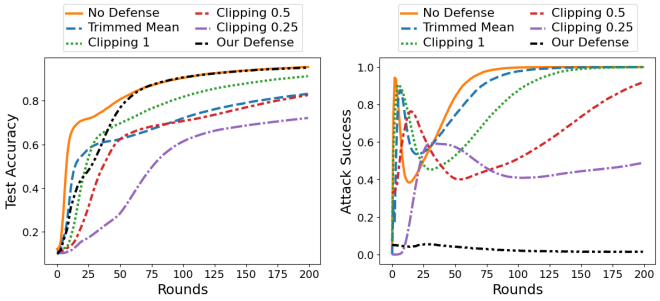
Robust Aggregation. We adapt Trimmed Mean [48] to P2PFL, by sorting all the peer updates and filtering out the p highest and lowest values and averaging the remaining updates. In our experiments (Figure 9), we observed for $p = 1$ that Trimmed Mean is not effective against the backdoor attacks in P2PFL, as the attacker still achieves 100% attack success. Additionally, the learning procedure is slowed down, and accuracy converges slower than the "No Defense" case.

Clipping Defense. Gradient clipping represents another standard defense against poisoning attacks in FL. The most devastating poisoning attack in FL is model poisoning, where the contribution of each compromised node is amplified by the boosting factor applied to the local model. In the extreme, a model boosting attack could overwrite the global model, and therefore gradient clipping is critical for limiting the contribution of individual clients. In gradient clipping in FL, the server bounds the update sent by each participant by a threshold norm C before aggregation. We adapt this defense to the P2PFL setting that does not rely on a trusted server to aggregate and bound updates. Instead, each peer rescales all updates which contribute to its model using:

$$U^t_{j,C} = U^t_j / \max(1, ||U^t_j||/C) \quad (3)$$

U^t_j is the update sent by peer j at round t, $||U^t_j||$ is the ℓ_2 norm of the peer update, and C is the clipping norm. Selecting the clipping norm C is not straightforward, as there is a tradeoff between attack success and test accuracy. A large C reduces the impact of the defense, while a small C reduces the test accuracy. A node can generally trust its own updates, but bounding only neighbors' contributions and not its own offsets the benefit of using P2PFL instead of local training. Similarly, setting the clipping norm too small will reduce the benefit of aggregating updates from neighbors. On the other hand, a large norm enables potential attacks to be aggregated into the model.

We implemented our framework in Python's deep learning API Keras using the Adam optimizer — a stochastic gradient descent method based on adaptive estimation of first-order and second-order gradients. For each peer, we extract the weights of the current model, rescale them to fit within the clipped norm, and then apply the rescaled weights to update the model. To limit the contribution of malicious peers in P2PFL we first experimented with small clipping values (0.05), but the model did not converge. Next, we selected clipping norm values of 0.25, 0.5, and 1 and present results with these clipping norms in Figure 9. The 0.25 norm reduces the attack success from 1 to 0.4 (Figure 9b) but at a high cost on accuracy. The larger norms impose a smaller cost on accuracy, as the attack picks up, approaching the "No Defense" success of 1.0.



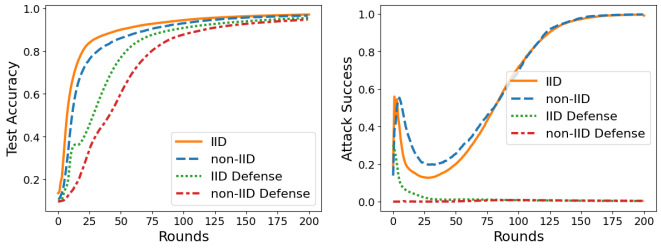
(a) Test Accuracy (b) Attack Success

Fig. 9. Defenses: (a) accuracy and (b) attack success on poisoned samples with 60-node Watts Strogatz topology, 10% adversarial nodes, PageRank strategy.

B. Our Defense

We showed in the previous section that standard gradient clipping (that uses a single norm for all participants) is ineffective in P2PFL settings. If the clipping norm for malicious peers is too large, the malicious updates will be aggregated into the local model. If the clipping norm for the local model is too small, the model's convergence is significantly impacted. Due to these two conflicting requirements of the clipping norm, we propose using two different clipping norm values, one for bounding the neighbor peers' updates and one for the local model. We have the flexibility to select a smaller norm for neighbor peers and a larger norm for the local model. We choose the neighboring norm as 0.1 and the local norm as 1, after experimenting with multiple values. In Figure 9 we study the effectiveness of using two separate clipping norms as a defense strategy against the PageRank-based poisoning attack.

We observe that after 200 rounds of training, the attack success is 0, while accuracy reaches 0.96.



(a) Test Accuracy (b) Attack Success

Fig. 10. Defenses in IID versus non-IID ($\alpha = 1$) settings: (a) accuracy and (b) attack success on poisoned samples. Configuration: 60-node Watts Strogatz topology, 5% adversarial nodes selected with the PageRank strategy.

While these experiments were carried out in IID settings, we also evaluated our defense with non-IID data distributions where the clients receive different numbers of samples per class (with $\alpha = 1$). These results are represented in Figure 10. As previously noted (Figure 7a), accuracy converges slower in non-IID even without defenses. The clipping process imposes an additional slowdown on accuracy convergence. However, our defense helps to correctly classify poisoned samples, and thus, significantly reduces the attack success in both IID and non-IID settings. Figure 10b shows that within 100 rounds of training, the attack is essentially stifled (success rate of 0).

Our two-norm defense is the only strategy we are aware of that is effective against P2PFL backdoor attacks, and it obtains better accuracy under attack than Trimmed Mean and gradient clipping. The main insight behind the defense is that the trusted local model of a node is given a higher weight, while the peer models are assigned lower weight, limiting their contribution to the node's final model. The defense can be extended by using different weights for neighboring peers, based on the level of trust a node has for each peer. We leave this as an exploration for future work, in addition to testing the impact of poisoning attacks and defenses in P2PFL using other datasets and model architectures.

VII. Related Work

We review related work in three areas: P2PFL algorithms, FL poisoning attacks, and emergent FL architectures.

Peer-to-peer Federated Learning. Federated Learning [6], [7] trains ML models collaboratively to preserve data privacy. Peer-to-Peer Federated Learning is a distributed learning paradigm that removes dependence on a trusted aggregation server. [20] proposed a fast algorithm for non-Byzantine settings, while several works address the Byzantine model where compromised nodes send arbitrary updates. [16] introduces *coordinate descent* that is robust against Byzantine failures but not scalable, while [15] designs a scalable algorithm for Byzantine faults. [12] introduces personalized Byzantine robust P2P learning for deep networks. [64] designs a clipping-based defense that assigns weights to neighbor contributions for aggregation and has provable convergence. None of these works study backdoor attacks in P2PFL.

Poisoning and Backdoor Attacks in FL. Poisoning attacks, including targeted and backdoor attacks have been extensively studied in classical ML [23]–[31], and in FL [34], [37]. In addition to standard poisoning attacks, recent work on network-level adversaries in FL showed that adversaries might cleverly drop network packets and significantly reduce the model's performance on sub-populations [65].

Edge Federated Learning. Recently, FL has been studied in the context of edge computing [66], [67]. Edge federated learning leverages data collected on widely dispersed edge devices, such as IoT and new 5G technologies to learn a global model shared by multiple decentralized edge clients.

VIII. Conclusion

We proposed a new modular P2PFL architecture that separates the learning and communication graphs, and allows to emulate a P2PFL system by instantiating real network topologies. We studied backdoors attacks in P2PFL and showed that a small number of attackers (5% of nodes) could achieve a high attack success, without decreasing the model's accuracy on clean data. We showed that defenses proposed for centralized FL settings, such as gradient clipping and Trimmed Mean, are ineffective in P2PFL. We propose a new defense that uses a weighted combination of the local model and model updates sent by a node's peer by assigning a higher weight to the trusted local model. Our work opens up new avenues for experimenting with other learning protocols in P2PFL architectures, and evaluating other attacks and defenses.

Acknowledgements

This research was supported by the Department of Defense Multidisciplinary Research Program of the University Research Initiative (MURI) under contract W911NF-21-1-0322.

References

[1] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.

[2] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM CSUR*, 2019.

[3] G. Manogaran and D. Lopez, "A survey of big data architectures and machine learning algorithms in healthcare," *International Journal of Biomedical Engineering and Technology*, vol. 25, pp. 182–211, 2017.

[4] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," *A Practical Guide, 1st Ed., Cham: Springer*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[5] E. L. Harding, J. J. Vanto, R. Clark, L. Hannah Ji, and S. C. Ainsworth, "Understanding the scope and impact of the California Consumer Privacy Act of 2018," *Journal of Data Protection & Privacy*, 2019.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.

[7] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party ML*, 2016.

[8] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *CCS*, 2017, pp. 1175–1191.

[9] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *ICLR*, 2018.

[10] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" in *Advances in Neural Information Processing Systems*, 2020.

[11] Y. Wen, J. Geiping, L. Fowl, M. Goldblum, and T. Goldstein, "Fishing for user data in large-batch federated learning via gradient magnification," in *ICML*, vol. 162, 2022, pp. 23 668–23 684.

[12] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Personalized and private peer-to-peer machine learning," in *AISTATS*, 2018, pp. 473–481.

[13] P. Vanhaesebrouck, A. Bellet, and M. Tommasi, "Decentralized collaborative learning of personalized models over networks," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 509–517.

[14] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," *arXiv preprint arXiv:1901.11173*, 2019.

[15] C. Fang, Z. Yang, and W. U. Bajwa, "Bridge: Byzantine-resilient decentralized gradient descent," *IEEE T Signal Information Processing*, 2022.

[16] Z. Yang and W. U. Bajwa, "Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning," *IEEE Trans. on Signal and Inform. Processing over Networks*, vol. 5, no. 4, pp. 611–627, 2019.

[17] K. Kuwaranancharoen, L. Xin, and S. Sundaram, "Byzantine-resilient distributed optimization of multi-dimensional functions," in *ACC*, 2020.

[18] J. Peng, W. Li, and Q. Ling, "Byzantine-robust decentralized stochastic optimization over static and time-varying networks," *Signal Processing*, vol. 183, p. 108020, 2021.

[19] N. Gupta and N. H. Vaidya, "Byzantine fault-tolerance in peer-to-peer distributed gradient-descent," *arXiv preprint arXiv:2101.12316*, 2021.

[20] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "d^2: Decentralized training over decentralized data," in *ICML*, 2018, pp. 4848–4856.

[21] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, and C. Parada, "Personalized speech recognition on mobile devices," *arXiv*, 2016.

[22] F. Granqvist, M. Seigel, R. van Dalen, A. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *arXiv*, 2020.

[23] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[24] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *AAAI*, 2015.

[25] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?" in *ICML*, 2015.

[26] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *ICML*. PMLR, 2017, pp. 1885–1894.

[27] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv*, 2017.

[28] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning FAIL? generalized transferability for evasion and poisoning attacks," in *USENIX Security*, 2018, pp. 1299–1316.

[29] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *NeurIPS*, vol. 31, 2018.

[30] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *S&P*. IEEE, 2018, pp. 19–35.

[31] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, 2019.

[32] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *ESORICS*. Springer, 2020.

[33] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Model poisoning attacks in federated learning," in *SecML*, 2018, pp. 1–23.

[34] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *AISTATS*. PMLR, 2020.

[35] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *USENIX Security*, 2020.

[36] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[37] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.

[38] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *NeurIPS*, vol. 33, 2020.

[39] F. V. Martins, E. G. Carrano, E. F. Wanner, R. H. Takahashi, and G. R. Mateus, "A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks," *IEEE Sensors*, 2010.

[40] Y. Jiang, X. Ge, Y. Zhong, G. Mao, and Y. Li, "A new small-world IoT routing mechanism based on Cayley graphs," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 384–10 395, 2019.

[41] Z. Dong, Z. Wang, W. Xie, O. Emelumadu, C. Lin, and R. Rojas-Cessa, "An experimental study of small world network model for wireless networks," in *IEEE Sarnoff Symposium*, 2015, pp. 70–75.

[42] C. Liu and G. Cao, "Distributed critical location coverage in wireless sensor networks with lifetime constraint," in *INFOCOM*. IEEE, 2012.

[43] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv*, 2017.

[44] D. Vyzovitis, Y. Napora, D. McCormick, D. Dias, and Y. Psaras, "GossipSub: Attack-Resilient Message Propagation in the Filecoin and ETH2.0 Networks," *arXiv*, 2020.

[45] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[46] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[47] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *ICML*. ACM, 2018, pp. 3521–3530.

[48] C. Xie, O. Koyejo, and I. Gupta, "Phocas: dimensional byzantine-resilient stochastic gradient descent," *arXiv*, 2018.

[49] S. P. Borgatti, "Structural holes: Unpacking Burt's redundancy measures," *Connections*, vol. 20, no. 1, pp. 35–38, 1997.

[50] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[51] J. Saramäki, M. Kivelä, J.-P. Onnela, K. Kaski, and J. Kertesz, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review E*, vol. 75, no. 2, p. 027105, 2007.

[52] A. Chernikova, N. Gozzi, S. Boboila, P. Angadi, J. Loughner, M. Wilden, N. Perra, T. Eliassi-Rad, and A. Oprea, "Cyber network resilience against self-propagating malware attacks," in *ESORICS*. Springer, 2022.

[53] P. Erdos, A. Rényi *et al.*, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[54] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[55] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[56] M. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[57] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," 2017.

[58] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," 2017.

[59] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, 2012.

[60] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *ICDE*. IEEE, 2022, pp. 965–978.

[61] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *ICML*, vol. 97. PMLR, 2019, pp. 7252–7261.

[62] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *NIPS*. ACM, 2020.

[63] S. Hong, V. Chandrasekaran, Y. Kaya, T. Dumitraş, and N. Papernot, "On the effectiveness of mitigating data poisoning attacks with gradient shaping," *arXiv*, 2020.

[64] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust decentralized learning via ClippedGossip," 2023.

[65] G. Severi, M. Jagielski, G. Yar, Y. Wang, A. Oprea, and C. Nita-Rotaru, "Network-level adversaries in federated learning," in *CNS*. IEEE, 2022.

[66] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE COMMAG*, vol. 5, 2020.

[67] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, 2020.