Network-Level Adversaries in Federated Learning

Cristina Nita-Rotaru
Khoury College of Computer Science

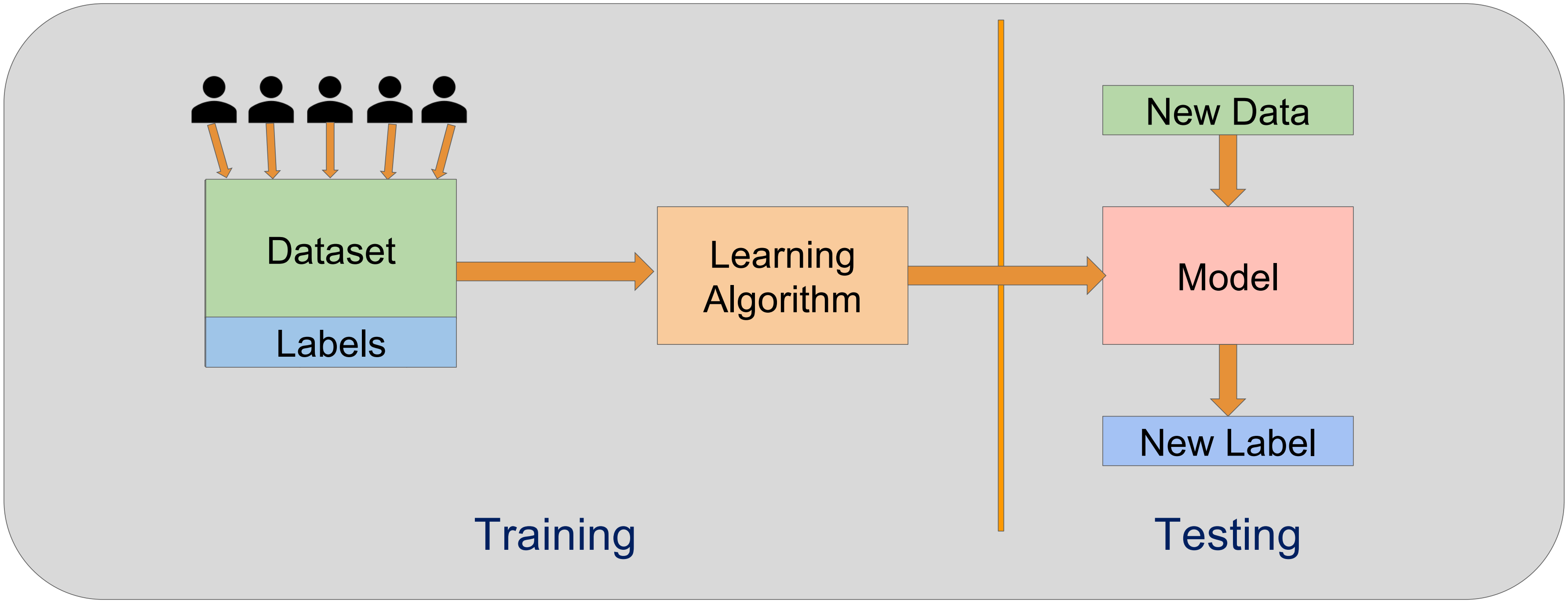
CROSSING Conference 2023

Acknowledgments

- Supported by the Department of Defense Multidisciplinary Research Program of the University Research Initiative (MURI) under contract W911NF-21-1-0322.
- Joint work with my colleague Alina Oprea, former students Matthew Jagielski (PhD) currently with Google Brain, Yuxuan (Ewen) Wang (undergrad) currently with Linkedin, and current students Giorgio Severi and Gokberg Yar, and research scientist Simona Boboila

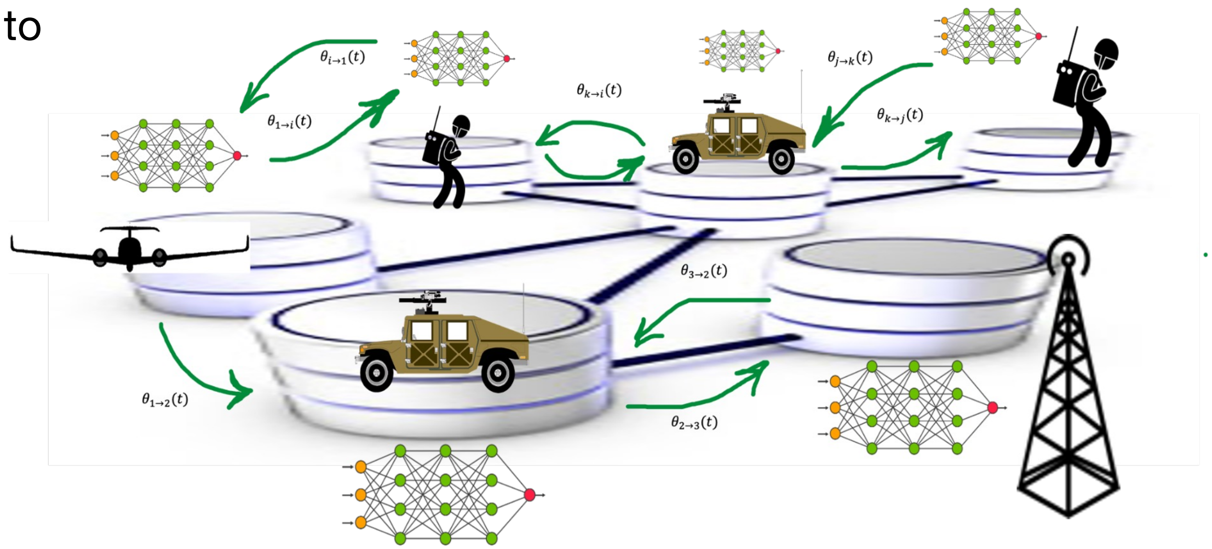
- **Network-Level Adversaries in Federated Learning**. Giorgio Severi, Matthew Jagielski, Gokberg Yar, Yuxuan Wang, Cristina Nita-Rotaru, Alina Oprea. In IEEE CNS 2022.
- **Backdoor Attacks in Peer-to-Peer Federated Learning.** Gokberk Yar, Simona Boboila, Cristina Nita-Rotaru, Alina Oprea. IEEE CNS 2023.

Supervised Machine Learning



Need for Collaborative Learning

- **Insufficient observations**
  - Data at a node is insufficient to learn a good model
- **Storage constraints**
  - Data too big to collect in one place
- **Limited computation**
  - Can not perform all computation in one place
- **Privacy concerns**
  - Data must remain where it was generated



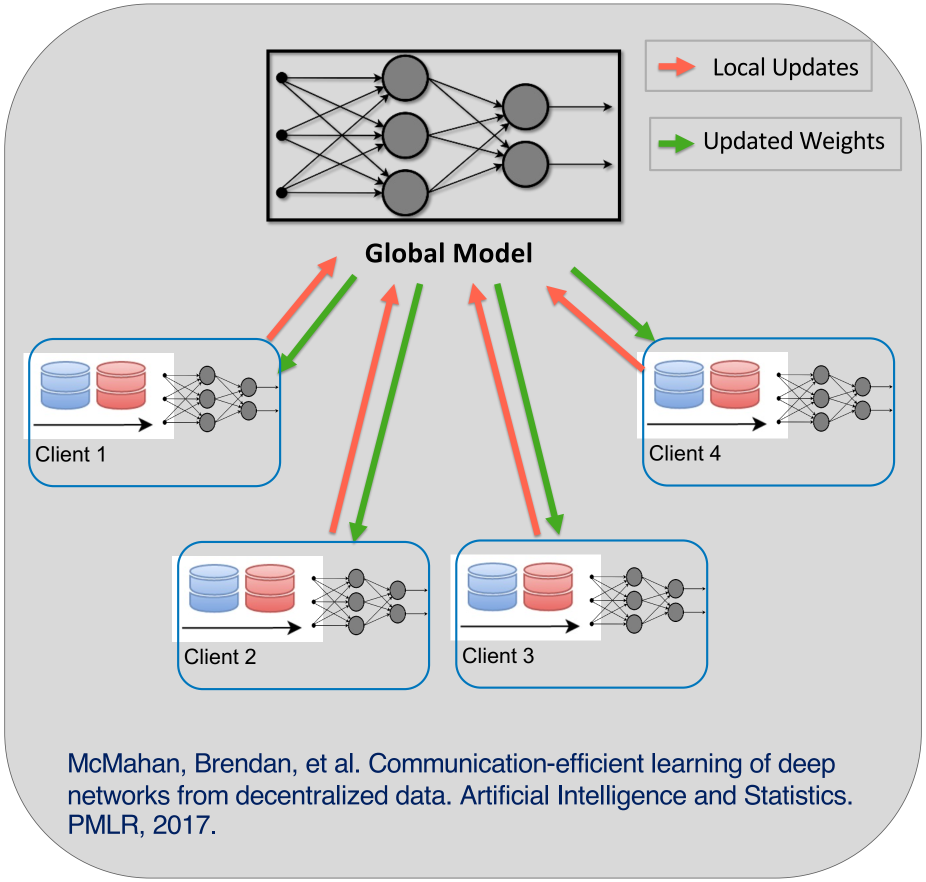
A better model can be learned if entities collaborate!

Federated Learning

- Clients train a machine learning model with the help of an aggregation server
  - Federated Averaging [McMahan et al. 2017]

- Training is an iterative process
  - Clients receives global model
  - Subset of clients update the model using local data and send updates to server
  - Server updates the global model by aggregating client contributions

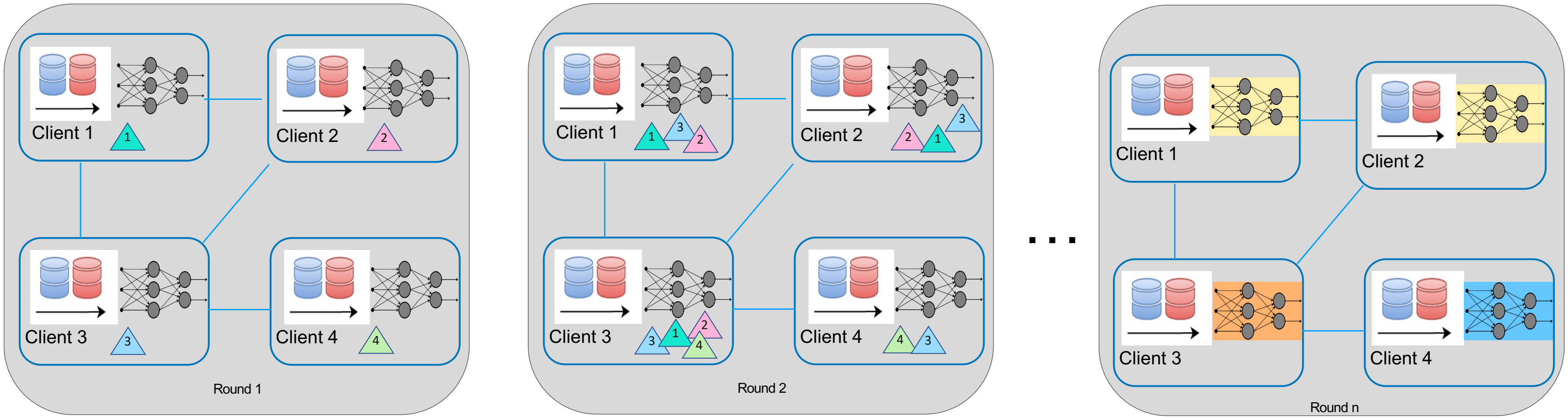
- Benefits
  - Training data remains on client devices
  - Computational efficiency



McMahan, Brendan, et al. Communication-efficient learning of deep networks from decentralized data. Artificial Intelligence and Statistics. PMLR, 2017.

P2P Federated Learning

- **No central server**
  - Clients (peers) collaborate to learn a personal or global model



Goals for Machine Learning

- **Accuracy**
- **Precision**
- **Recall**
- **F-score**
- **MSE**

- **Robustness**
  - Algorithms should be resilient to changes when using it on new data vs the training dataset
- **Fairness**
  - Datasets the models are trained on should be representative and avoid biases
- **Privacy**
  - Use of model should not reveal information about data it was trained on
- **Security**
  - Models should work correctly in the presence of attacks

Attacks against Machine Learning

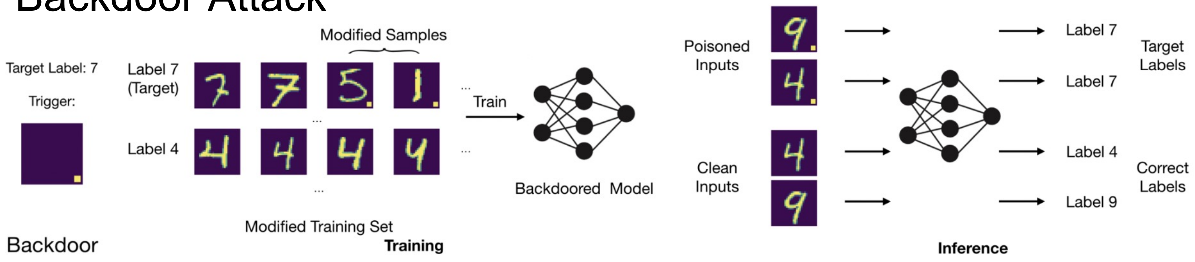
- **Attacker’s objective**
  - Targets system goals
- **Attacker’s capability**
  - Resources available
  - How/when interacts with the system: inject/change data/model updates
- **Attacker’s knowledge**
  - What they know

Attacker’s Objective

Learning stage

	Targeted Target small set of points	**Availability** Target majority of points	**Privacy** Learn sensitive information
Training	Targeted Poisoning Backdoor Trojan Attacks	Poisoning Availability Model Poisoning	-
Testing	Evasion Attacks Adversarial Examples	-	Reconstruction Membership Inference Model Extraction

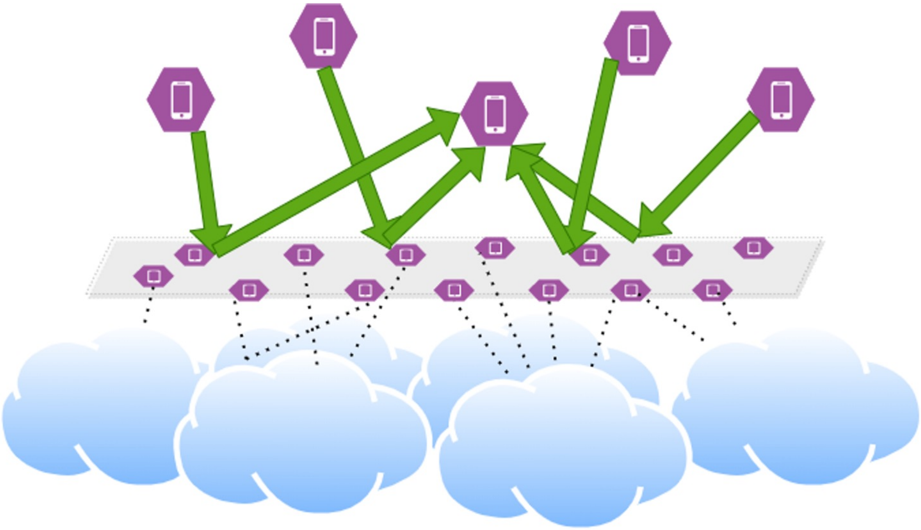
Backdoor Attack



Network-level Attacks

- **Can impact communication directly between two parties**
  - Modify
  - Inject
  - Replay
  - Drop

- **Can impact communication indirectly by influencing the communication graph**
  - Partition the network
  - Disconnect clients
  - Disconnect servers



For federated learning communication is: data, local updates, updated model

Network-level Attacks against ML: Challenges

- **Attacker's goal:**
  - Preventing or changing communication to impact model accuracy

- **Attacker's capability**
  - Network partitions difficult to create and detectable
  - Cryptography can prevent modification/injection
  - Attack must be sustained as machine learning is iterative

- **Attacker's knowledge**
  - Global network information is difficult to obtain
  - Channels can be encrypted

It is not clear how effective network-level attacks would be against machine learning!

In this talk

FEDEREATED LEARNING

Can an adversary with network-level capability decrease the accuracy of the machine learning models?

Can an adversary with network-level capability further amplify their attack with poisoning attacks?

Can we mitigate network-level attacks?

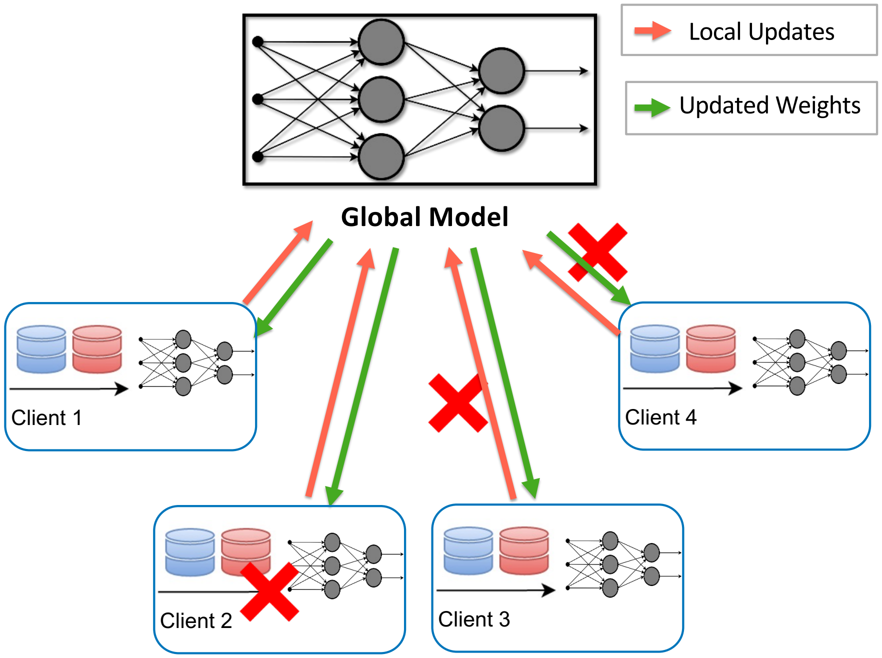
Federated Learning: Attacker Model

- **Attacker Goal**
  - Reduce accuracy on a target class

- **Attacker Capabilities**
  - Attacker can interfere with the delivery of model updates or the global model weights
  - Attack can be amplified by compromising a few clients and modifying their updates

- **Attacker Knowledge**
  - Attacker has access to global model in each round. (*Federated learning is an open system, the attacker can participate as one of the clients.*)

$$f_t = f_{t-1} + \eta \frac{\sum_{i=1}^{m} U_i}{m}$$



Attacker Capabilities

Dropping attacks

- The attacker drops data for a subset of clients and prevent their updates from getting to the server
    - **Random dropping**: Selection of random victim clients and dropping their messages
    - **Targeted dropping**: Identifying clients whose updates contribute significantly to the target class

Model poisoning attacks

- The attacker compromises a subset of clients
- They can send malicious updates by performing model poisoning attacks
- Previous work studied model poisoning in isolation [Bagdasaryan et al. 2020, Bhagoji et al. 2019], but we are interested in amplifying network-level attacks

Network Attack Model

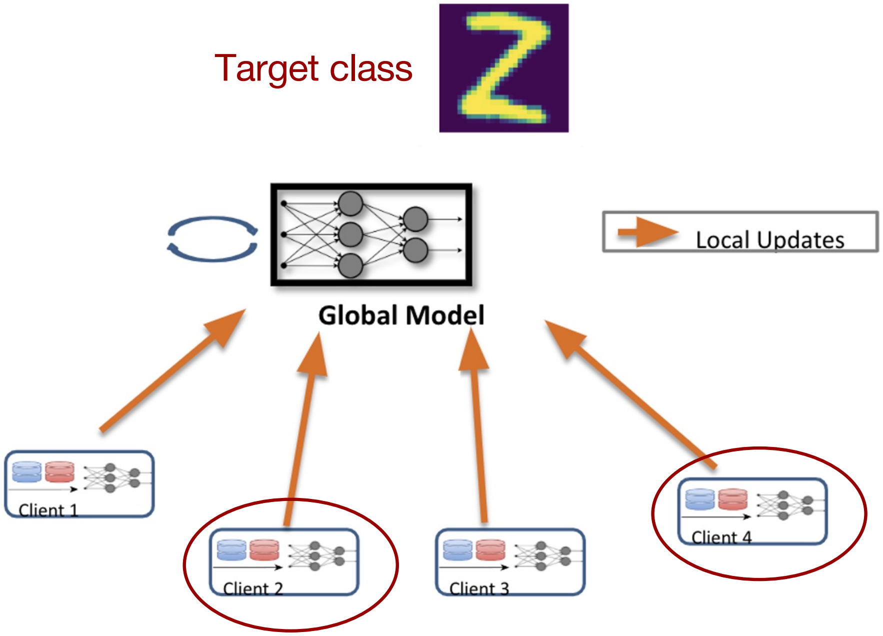
- **COMM_PLAIN**: All communication between clients and server is unencrypted
  - Network-level adversary obtains maximum information, as they can observe all the transmitted data
  - Most powerful adversary, useful for evaluating defenses

- **COMM_ENC**: All communication between clients and server is encrypted
  - Network-level adversary could infer the set of clients participating in each round, but not the exact model updates they send

- **COMM_ENC_LIMITED**: All communication is encrypted, and adversary observes only a subset of clients (has limited visibility)
  - Most constrained and realistic adversary

Methodology: Dropping Attacks

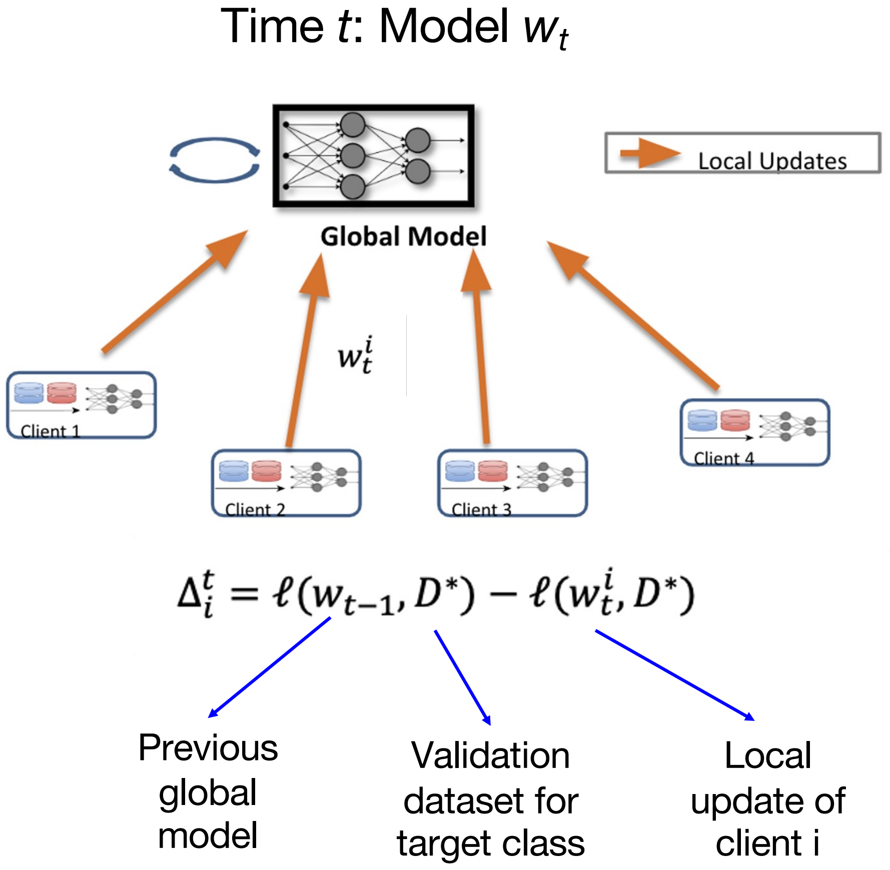
- **Naive strategy**: Drop updates from randomly selected clients
- **Main observation**: Clients do not contribute equally to the target class
  - Data is non-iid in FL deployments [Kairouz et al. 2019]
- **Insight**: Design **Client Identification** method to identify the top performing clients for target class
  - Observe client updates for a number of rounds before dropping
  - Results in more effective targeted dropping strategy



Clients 2 and 4 contribute to target class

Client Identification: COMM_PLAIN

- **COMM_PLAIN**: The attacker can observe individual clients' updates
- **Strategy: compute the model loss difference before and after a client's update on target class**
  - Repeat for multiple rounds
  - Rank clients by largest difference in loss across rounds
- Challenge: How to handle encrypted communication?



Time t: Model w_t

$$\Delta_i^t = \ell(w_{t-1}, D^*) - \ell(w_t^i, D^*)$$

Previous global model

Validation dataset for target class

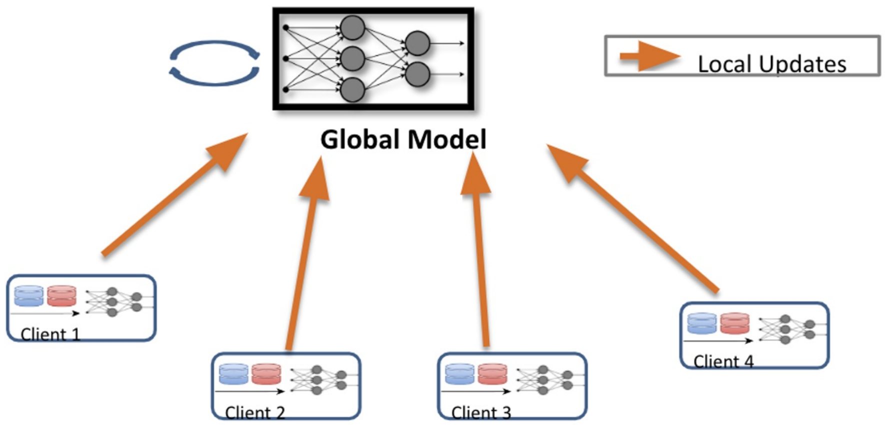
Local update of client i

Client Identification: COMM_ENC

- **COMM_ENC**: The attacker cannot observe individual clients' updates, only global model aggregates

- **Strategy: compute the model loss difference using the previous and current aggregated global models**
  - Apply the loss difference to all participating clients in current round
  - Repeat for multiple rounds
  - Rank clients by largest difference in loss across rounds
  - Requires more observation rounds than **COMM_PLAIN**

Time t: Model w_t



$$\Delta_i^t = \ell(w_{t-1}, D^*) - \ell(w_t, D^*)$$

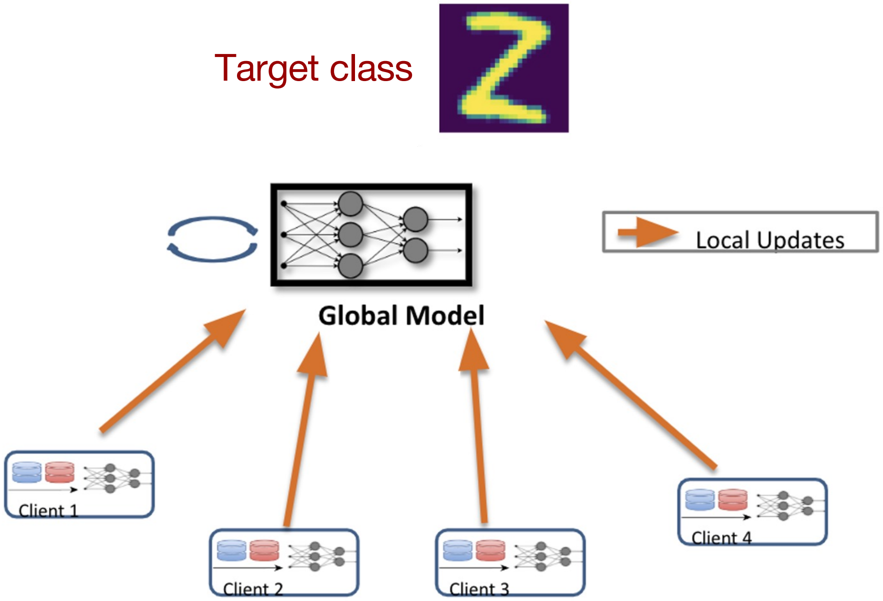
Previous global model

Current global model

Client Identification: Parameters

- **How many clients to drop?**
  - Tradeoff between attack success on target class and remaining stealthy on other data

- **How many observation rounds are needed to identify top clients?**
  - Wait number of rounds so that all clients of interest are observed at least once
  - Use coupon collector for analysis: O(n/m log n), where n is total number of clients and m is number of clients sampled per round



Evaluation Setup

Dataset / Modality	Task/Classes	Model	FL Parameters
EMNIST Images	Digit recognition 10	CNN	100 clients 1000 samples each
FashionMNIST Images	Image recognition 10	CNN	60 clients 400 samples each
DBPedia Text	Text classification 14	GloVE embedings and one-dimensional CNN	60 clients 1000 samples each

Target distribution

- One of the classes in the dataset (0, 1, or 9)
- Assume k clients have examples from target class, k = {9,12,15}
- Heterogeneous data: the k clients have 50% of examples from target class, and the rest are sampled with Dirichlet distribution

Client Identification Results

Average number of identified clients for target class 0, k=15 clients

Network Communication	Dataset	T=5	T=10	T=15	T=20	T= 50	T=70
COMM_PLAIN	EMNIST	4.25	9.5	11.5	12.0	14.0	14.0
	DBPedia	8.0	13.25	13.75	15.0	15.0	15.0
COMM_ENC	EMNIST	3.0	4.0	4.0	3.75	5.75	7.0
	DBPedia	5.25	7.0	8.0	9.0	11.25	11.75

- Findings:
  - Under **COMM_PLAIN** all clients are identified for DBPedia after 20 rounds
  - Fewer clients identified under **COMM_ENC**, but still on average more than 2/3 of clients are identified for DBPedia after 50 rounds
  - Number of rounds for convergence is 100 for EMNIST and 200 for DBPedia

Targeted Dropping for COMM_PLAIN

Accuracy on target class for k=15 clients

Dataset	Acc	Dropped k/3		Dropped 2k/3		Dropped k	
		Random	Targeted	Random	Targeted	Random	Targeted
EMNIST	0.80	**0.82**	**0.74**	**0.81**	**0.50**	**0.82**	**0.02**
FashinMNIST	0.55	**0.53**	**0.23**	**0.53**	**0.03**	**0.5**	**0.00**
DBPedia	0.53	**0.54**	**0.01**	**0.47**	**0.00**	**0.45**	**0.00**

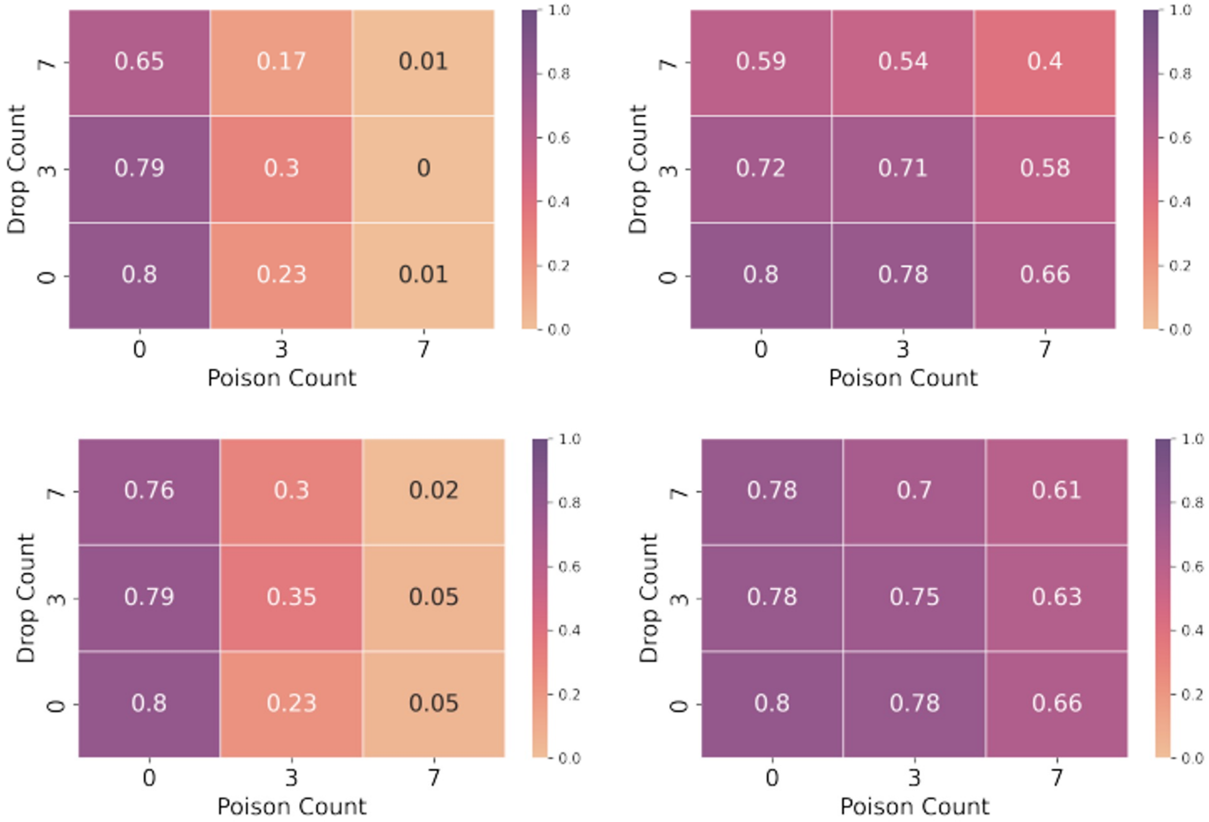
- Baseline: randomly drop the same number of clients
- For some datasets (DBPedia). targeted dropping of k/3 clients is catastrophic
- Overall model accuracy remains similar to original before attack
- Results are similar for **COMM_ENC**: k/3 dropping results in 0.38 accuracy on FashionMNIST and 0.06 accuracy on DBPedia

Targeted Dropping and Model Poisoning

T=100

COMM_PLAIN

COMM_ENC



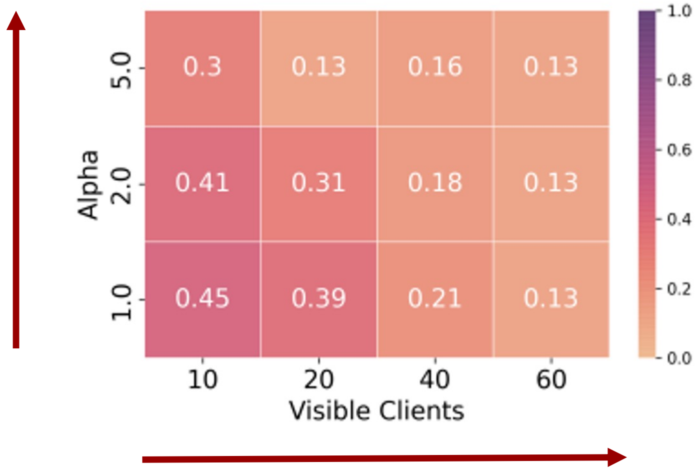
FedAvg, No clipping

FedAvg, with Clipping

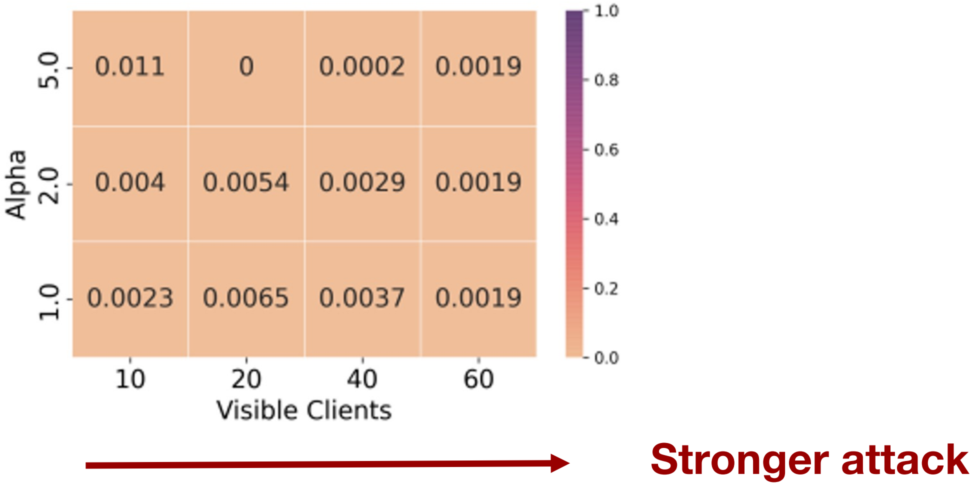
Use backdoor attack [Gu et al. 2017] with model poisoning [Bagdasaryan et al. 2020], [Sun et al. 2019]

Limited Adversarial Visibility

Drop



Drop and Poison



- **COMM_ENC_LIMITED**
  - Adversary observes a subset of clients (between 10 and 60 on x axis)
  - Parameter alpha (y axis) controls how many clients from the observed subset are from the target distribution
- **Attack is successful even under limited visibility!**

In this talk

Can an adversary with network observability capability influence the machine learning model in federated learning?

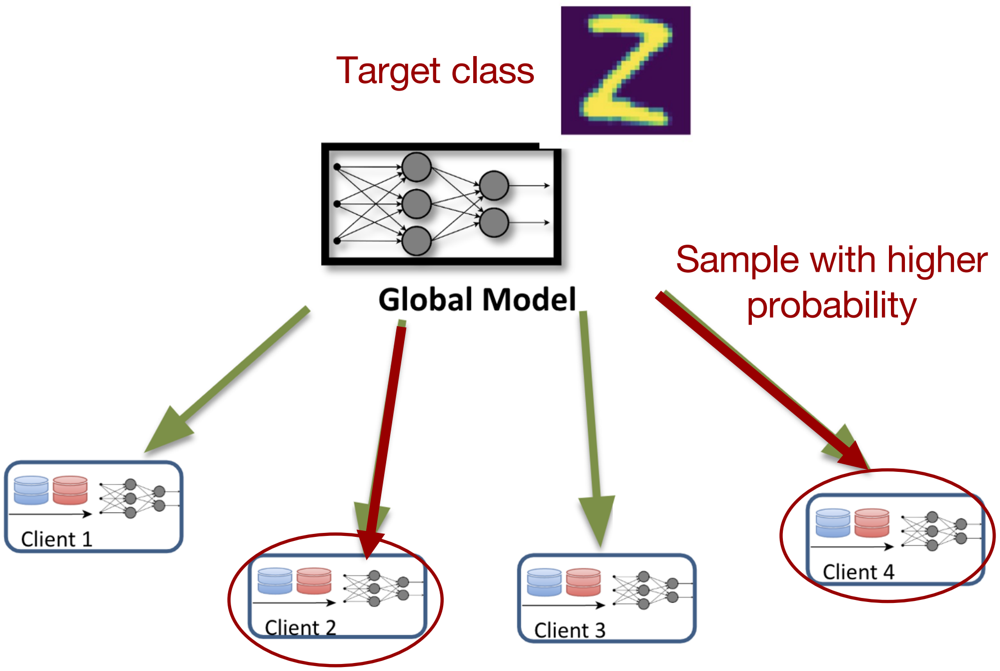
Can an adversary with network observability capability amplify his attack?

Can we mitigate network-level attacks in federated learning?

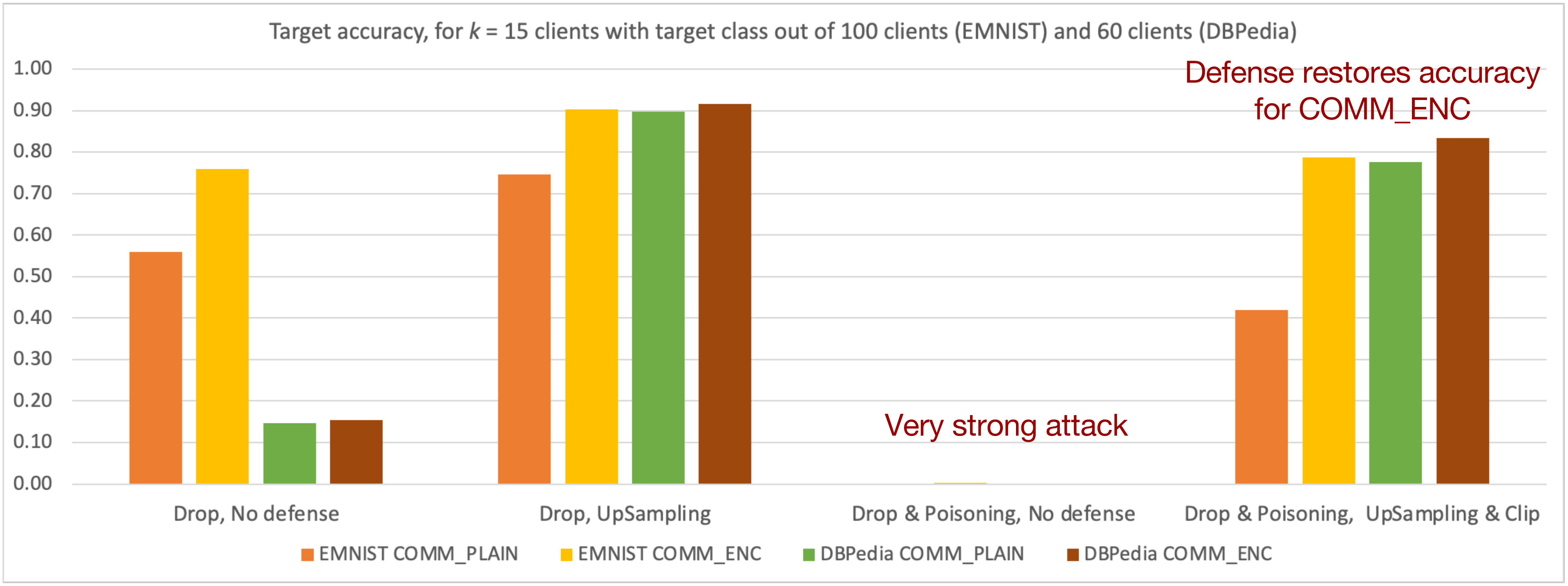
Defense: UpSampling

- **Key insight**
  - Use the same Client Identification procedure to identify important clients for the target class
  - Server knowledge
    - Individual client models under both **COMM_PLAIN** and **COMM_ENC**
    - Aggregated models under MPC

- **UpSampling Defense**
  - Server runs Client Identification to rank clients
  - Increase sampling weight proportional to rank
  - Can be combined with network-level defenses [Awerbuch et al. 2008, Obenshain et al. 2016] and poisoning defenses, e.g., gradient clipping [Sun et al. 2019]



Defense Evaluation



Defense restores accuracy for COMM_ENC

Very strong attack

In this talk

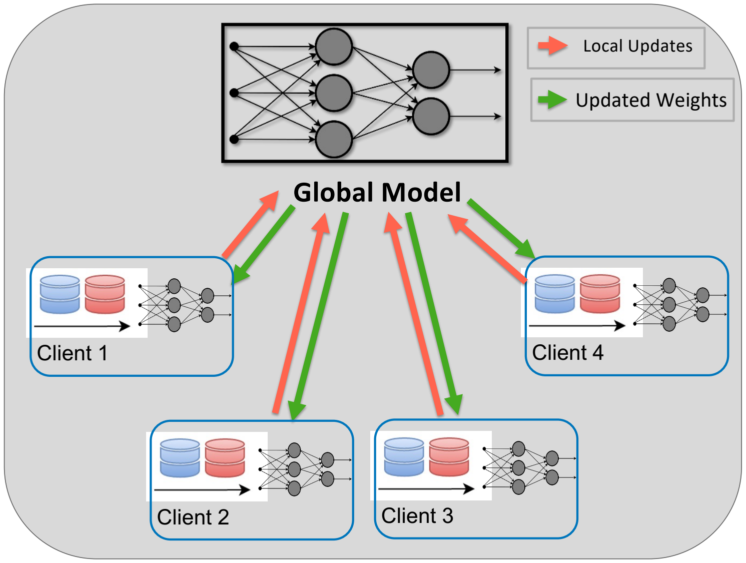
Can an adversary with network observability capability influence the machine learning model in federated learning?

Can an adversary with network observability capability amplify his attack?

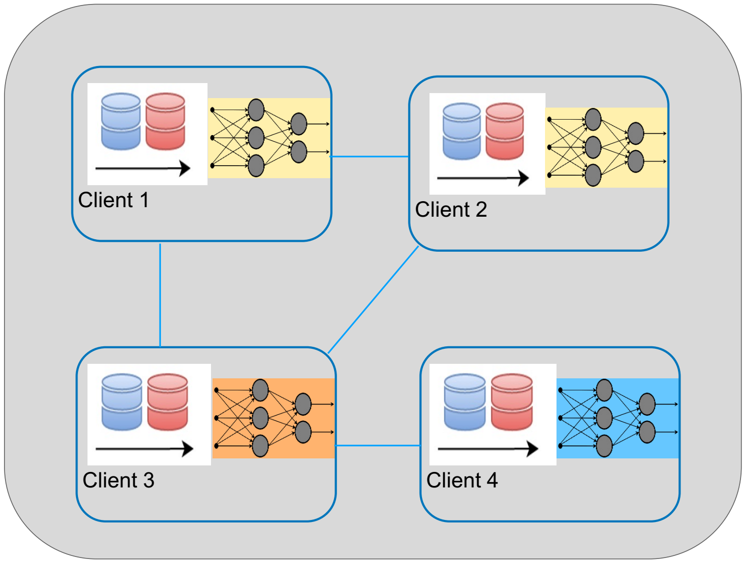
Can we mitigate network-level attacks in federated learning?

What about attacks against P2P federated learning ?

Centralized vs P2P Federated Learning



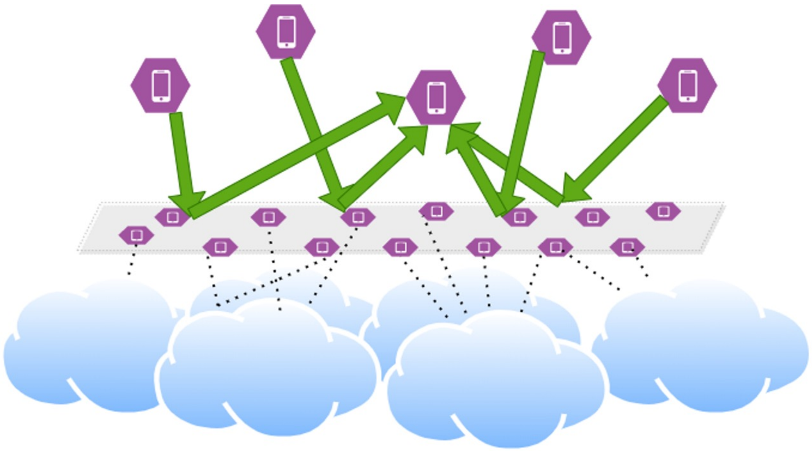
- Server acts as central aggregator
- Server is a single point of failure
- Asymmetric resources
- Communication is point to point
- Learning is through central aggregator



- No central trusted aggregator
- No single point of failure
- Symmetric resources at each peer
- Communication is multi-hop
- Learning is through a graph

Communication and Learning Graphs

- **Communication network**: failures, partitions
  - Physical network: real network
  - Logical network: overlay, may share physical links
- **Learning network**: bootstrapping, convergence
  - Input peers: peers each node is learning from
  - Output peers: peers a node shares its model with

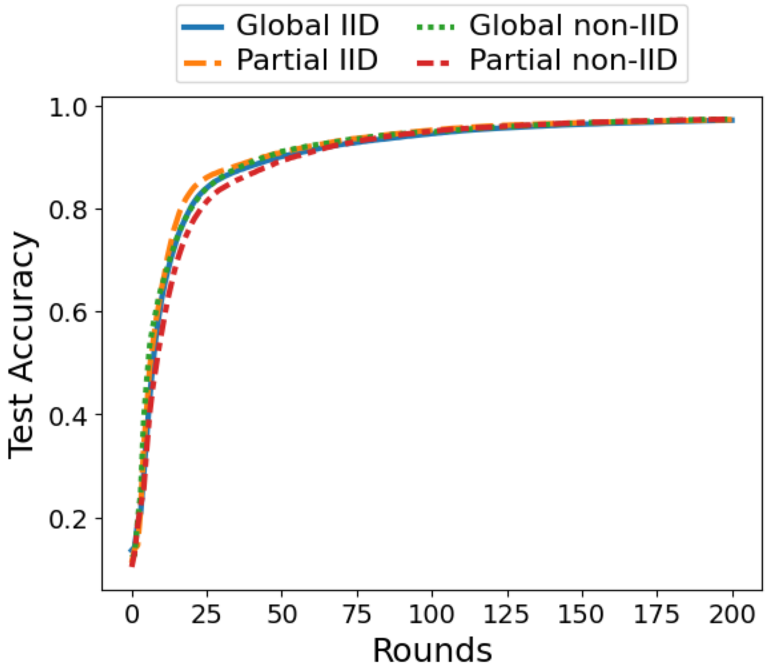


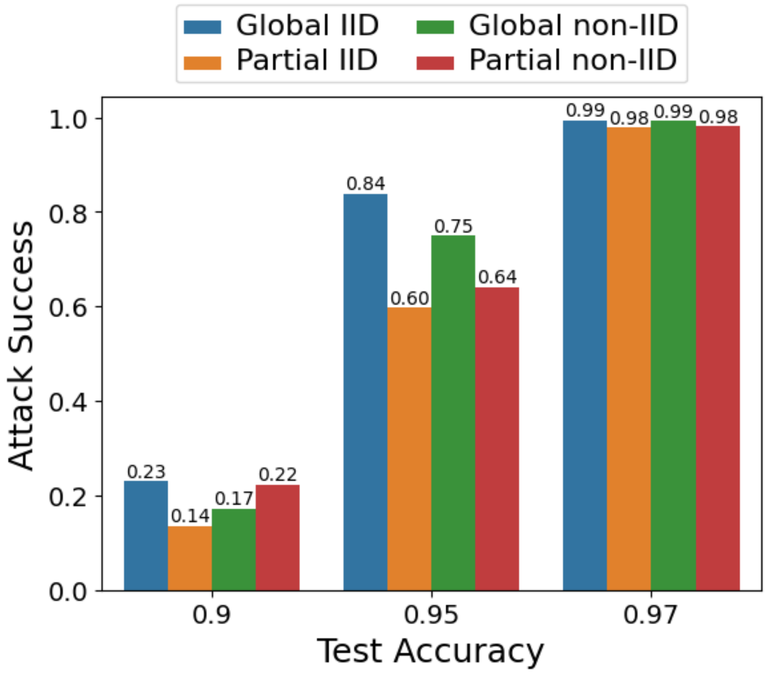
Dependencies and trade-offs between the learning network and the communication network

Backdoor attacks in P2P FL

Attacker has partial view of the P2P network

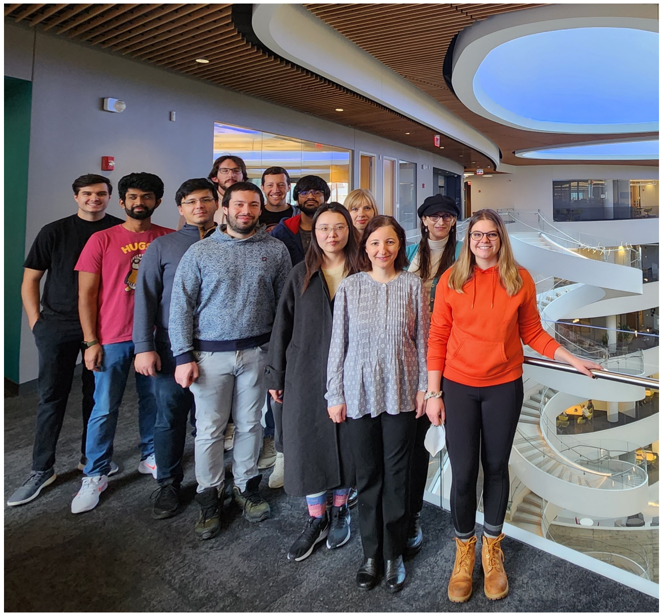
Graph: Watts Strogatz
Dataset: Emnist
Total: 60 nodes
Attackers:
3 nodes, PageRank
observes 20%nodes





Summary

- Showed that network-level attacks can impact accuracy of machine learning
    - Client Identification method that ranks clients by their contributions to the target class
    - Attack effective even when communication is encrypted and attacker can observe only a subset of clients
- Proposed UpSampling defense that modifies server-side sampling
- Performed evaluation on multiple datasets from image and text modalities
- Show attack is effective against P2PFL with partial graph observability



NDS2 Lab, Nov. 2022