Security & Robustness of Federated Learning

Yingfei Fan (yf2549), Yin Zhang (yz4053)
COMS6998-12 Midterm Seminar 10/26/2021

Content

- Motivation
- Background Work
- Adversarial Attacks on Model Performance
    - Paper 1
    - Paper 2
    - Observation and Insights
- Non-Malicious Failure Modes
    - Paper 3
    - Paper 4
    - Observation and Insights
- Conclusion

Motivation

Federated Learning: collaborative machine learning without centralized training data

- users' device: store data and train model
- central server on cloud: update model and send it back to clients

Two problems:

1. Federated learning is particularly susceptible to non-malicious failures from unreliable clients outside the control of the service provider
2. Federated learning systems are vulnerable to attacks from malicious clients.

Background Work

A bit introduction:

- System Design https://arxiv.org/abs/1902.01046: local data on Android devices + Tensorflow on the cloud => Application: Gboard
- Federated Averaging algorithm https://arxiv.org/abs/1602.05629:
  - motivated by bandwidth and latency limitations
  - can train deep networks using 10-100x less communication compared to a naively federated version of SGD
- Secure Aggregation protocol http://eprint.iacr.org/2017/281:
  - decrypt the average update if 100s or 1000s of users have participated
  - no individual phone's update can be inspected before averaging
- Compressing updates https://arxiv.org/abs/1610.05492: use random rotations and quantization to reduce upload communication costs
-
- Advances and Open Problems in Federated Learning https://arxiv.org/abs/1912.04977
  - Defending Against Attacks and Failures
  - non-malicious failures: noisy training labels, unreliable clients, unreliable communication
  - malicious failures: explicit attacks that target training and deployment pipelines

Background Work

Yingfei: Adversarial Attacks on Model Performance

- **data poisoning** (training-time attacks)   eg. flipping the labels <= Defense: model filtering
  https://link.springer.com/chapter/10.1007/978-3-030-58951-6_24
- **model update poisoning** : eg. Byzantine attacks <= Defense: replace the averaging step on the server with a robust estimate of the mean
  https://www.usenix.org/conference/usenixsecurity20/presentation/fang
- **evasion attacks** (inference-time attacks) eg. inserting adversarial examples <= adversarial training
  https://arxiv.org/abs/1708.06131

Yin: Non-malicious failures

- **Data pipeline failures** <= Solution: GENERATIVE MODELS
- **Noisy model updates (effects of noisy data)** <= Solution: Robust Design Under Expectation-Based/Worst-case Mode
- ***Client reporting failures**

Paper 1: **Attack of the Tails: Yes, You Really Can Backdoor Federated Learning (arix.org July 2020)**

What is blackdoor?

The goal of a backdoor, is to corrupt the global FL model into a targeted mis-prediction on a specific subtask, e.g., by forcing an image classifier to mis-classify green cars as frogs.

Contribution:

- Establish theoretically that if a model is vulnerable to adversarial examples, then, backdoor attacks are unavoidable (Detecting backdoors in a model is NP-hard)
- Invent a new family of backdoor attacks: edge-case backdoors (live on the tail of the input distribution)
- One can insert them across a range of machine learning tasks (e.g., image classification, OCR, text prediction, sentiment analysis)
- Robust to defense mechanisms based on differential privacy, norm clipping, and robust aggregators such as Krum and Multi-Krum

Definition 1. *Let $X \sim P_X$. A set of labeled examples $\mathcal{D}_{edge} = \{(\boldsymbol{x}_i, y_i)\}_i$ is called a p-edge-case examples set if $P_X(\boldsymbol{x}) \le p, \forall(\boldsymbol{x}, y) \in \mathcal{D}_{edge}$ for small $p > 0$.*

Adversarial Attacks on Model Performance

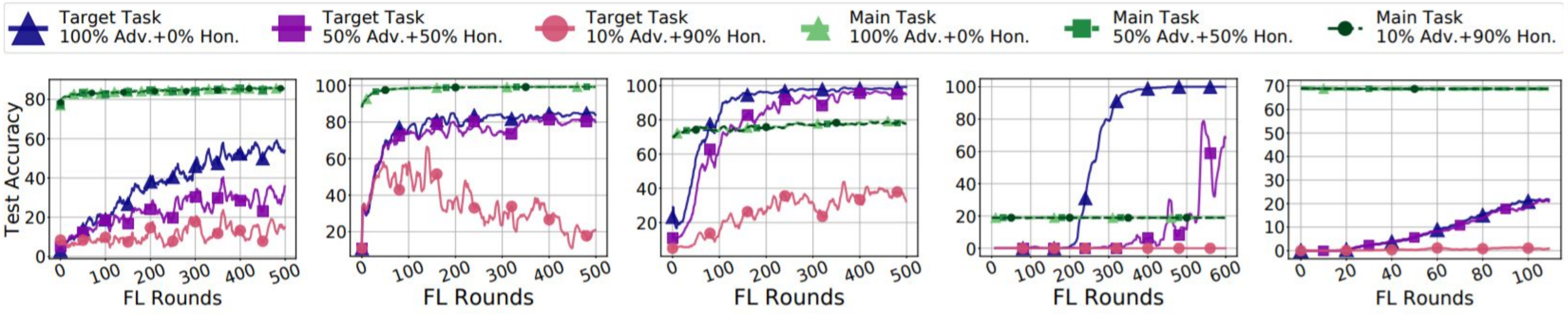
Paper 1: **Attack of the Tails: Yes, You Really Can Backdoor Federated Learning (arix.org July 2020)**

Constructing a p-edge-case example set

adversary: some mixture between D(benign samples) and D_edge(edge-case samples)

1. feed the DNN with benign samples
2. collect the output vectors of the penultimate layer
3. fit a Gaussian mixture model with the # of clusters = the # of classes **=>** we have a generative model with which the adversary can measure the probability density of any given sample and filter out if needed.

Experiments against state-of-the-art (SOTA) FL defenses -both black-box and PGD edge-case attacks



(a) Task 1 (b) Task 2 (c) Task 4 (d) Task 5 (e) Task 3

Paper 2: **FLGUARD: Secure and Private Federated Learning (axiv.org Jan 2021)**

Existing issue:

- No defense can protect the FL process against multi-backdoor attacks

Some details:

Existing defenses against backdoor attacks are based on two main ideas:

- model clustering for identifying potentially poisoned model updates
- differential privacy-based techniques such as clipping model weights and adding noise

Some observations:

- Existing clustering-based defenses aim to divide clients into $n = 2$ clusters: benign and malicious. If simultaneously injecting $m \geq n$ backdoors, such defense would not detect all of the attacks. **(too many attacks)**
- Adversary can evade any clustering approach by ensuring that the distance between poisoned model updates W' and benign models W remains smaller than the discriminative ability ε of the used clustering approach. **(attack models are too close to benign ones)**
- If the applied clipping bound α is too high, an adversary can boost its model W' by scaling up its weights up to the clipping bound, thereby maximizing the impact on the aggregated global model. **(clipping bound α cannot be too high)**
- If the applied clipping bound α is too low, also a large fraction of weights of benign model updates W will undergo clipping, thereby leading to a deterioration of the accuracy of the resulting aggregated global model on benign data. **(clipping bound α cannot be too low)**

Therefore, (1) Developing a new clustering approach capable of handling multiple simultaneous backdoors, (2) optimized parameter selection for clipping and noising, and (3) how to combine these approaches to achieve an effective defense.

- FLGUARD can entirely remove backdoors with a negligible effect on accuracy

Paper 2: **FLGUARD: Secure and Private Federated Learning (axiv.org Jan 2021)**

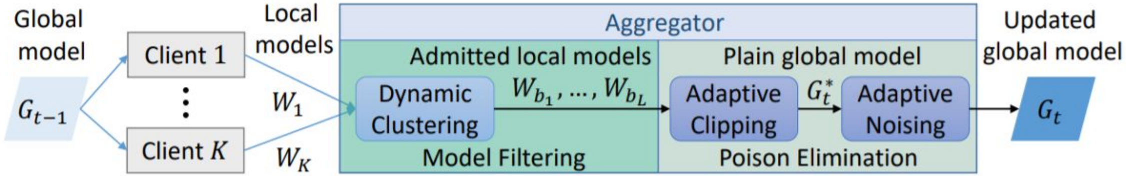


Fig. 1: Overview of FLGUARD in round t.

Dynamic Clustering:

- calculating the pairwise Cosine distances measuring the angular differences between all model updates
  - not affected by attacks that scale updates to boost their impact
- applying the HDBSCAN clustering algorithm
  - clusters the models based on their density and dynamically determines the required number of clusters

Poison Elimination :

L2-norms get smaller after each training iteration => uses adaptive clipping and noising

Defenses	Reddit		CIFAR-10		IoT-Traffic	
	TPR	TNR	TPR	TNR	TPR	TNR
Krum	9.1	0.0	8.2	0.0	24.2	0.0
FoolsGold	**100.0**	**100.0**	0.0	90.0	32.7	84.4
Auror	0.0	90.0	0.0	90.0	0.0	70.2
AFA	0.0	88.9	**100.0**	**100.0**	4.5	69.2
FLGUARD	22.2	**100.0**	23.8	86.2	**59.5**	**100.0**

Observation and Insights - Adversarial Attacks on Model Performance

- **Attacks and Defenses are always open questions**
- **Reality is the tradeoff between accuracy and robustness**

Doubts on the feasibility of fair and robust predictions by FL systems in their current form

Rethink how to guarantee robust and fair predictions in the presence of edge-case failures

Background Work

Yingfei: Adversarial Attacks on Model Performance

- **data poisoning** (training-time attacks)   eg. flipping the labels <= Defense: model filtering
  https://link.springer.com/chapter/10.1007/978-3-030-58951-6_24
- **model update poisoning** : eg. Byzantine attacks <= Defense: replace the averaging step on the server with a robust estimate of the mean
  https://www.usenix.org/conference/usenixsecurity20/presentation/fang
- **evasion attacks** (inference-time attacks) eg. inserting adversarial examples <= adversarial training
  https://arxiv.org/abs/1708.06131

Yin: Non-malicious failures

- **Data pipeline failures** <= Solution: GENERATIVE MODELS
- **Noisy model updates (effects of noisy data)** <= Solution: Robust Design Under Expectation-Based/Worst-case Mode
- ***Client reporting failures**

Non-Malicious Failure Modes - Noisy model updates

Paper 3: **Robust Federated Learning With Noisy Communication** (*IEEE* June 2020)

Problem Statement:

Noise problem in Federated Learning

=> Due to the noise existed during wireless communication, it is impractical to achieve perfect acquisition of the local models

=> Noise existed during communication process has serious effect on Federated Learning system performance

Goal: Improve the robustness of federated learning with noisy communication

Main contributions:

- Alleviate the effects of noise in the training process with a robust federated learning method
- Robust designs under 2 models:
  - **Expectation-based model** - based on the statistical properties of the noise uncertainty
  - **Worst-case model** - represents the fixed uncertainty sets of noise
  - Convergence analysis for the proposed design

SYSTEM MODEL:

- Distributed learning system consisting of a single central server and N edge nodes
- The training target is to minimize the global loss function *F*(w) according to the distributed learning i.e., $\mathbf{w}^* = \arg\min F(\mathbf{w})$.

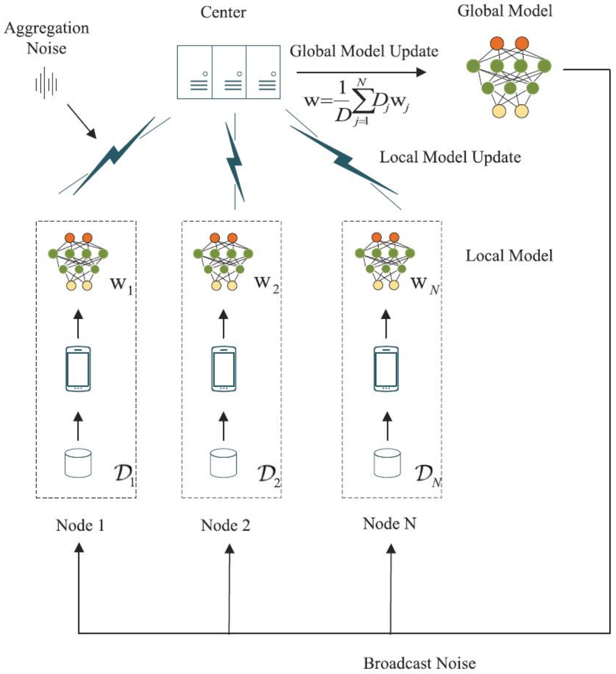


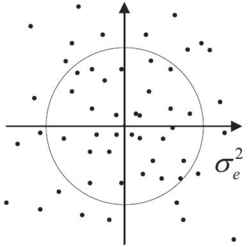
Fig. Federated learning with wireless communication

Paper 3: **Robust Federated Learning With Noisy Communication** (*IEEE* June 2020)

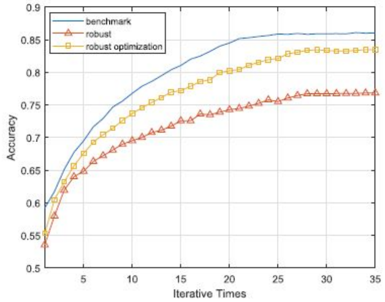
Effective noise as parallel optimization problem:

- **Expectation-based model**
  - SAM algorithm
  - Aims at optimizing either the long-term average performance or the outage performance
- **Worst-case model**
  - Sampling based SCA algorithm
  - Deterministic method to represent the instantaneous condition
  - Optimization is to find the local optimal model => min-max problem for each node

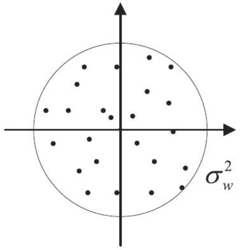
Convergence & Simulation Results: Both design methods can improve the prediction accuracy & the loss function values w/ acceptable convergence rates => Robust FL



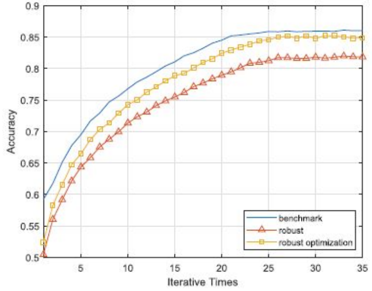
(a) Noise under expectation-based model in two-dimensional space.



(a) The accuracy performance versus iterative times under expectation-based model.



(b) Noise under worst-case model in two-dimensional space.



(a) The accuracy performance versus iterative times under worst-case model.

Fig. Noise under expectation-based model and worst-case model in 2-dimensional space

Fig. corresponding performance versus iterative times under two models

Paper 4: **Generative Models for Effective ML on Private, Decentralized Datasets** ***(ICLR 2020)***

Motivation:

- Potential data pipeline issues in federated learning:
    - Data restrictions makes detection significantly more challenging e.g. feature-level preprocessing issues
    - Raw data remains distributed across a fleet of devices, while an orchestrating server coordinates training of a shared global model
- Goal: Improve the robustness of federated learning with non-inspectable data

Task		Selection criteria for data to inspect
T1	Sanity checking data	Random training examples
T2	Debugging mistakes	Misclassified examples (by the primary classifier)
T3	Debugging unknown labels/classes, e.g. out-of-vocabulary words	Examples of the unknown labels/classes
T4	Debugging poor performance on certain classes/slices/users	Examples from the low-accuracy classes/slices/users
T5	Human labeling of examples	Unlabeled examples from the training distribution
T6	Detecting bias in the training data	Examples with high density in the serving distribution but low density in the training distribution.

Fig. ML modeler tasks typically accomplished via data inspection

Main Contributions:

- Identifying key challenges in implementing end-to-end workflows with non-inspectable data
- Propose a differentially private federated generative models that synthesize examples representative of the private data => resolve the challenges
- Demonstrated application of two example model classes (DP federated RNNs & GANs)
    - How privacy preserving federated generative models can be trained to high enough fidelity to discover introduced data errors matching those encountered in real world scenarios?

Non-Malicious Failure Modes - Data pipeline failures

Paper 4: **Generative Models for Effective ML on Private, Decentralized Datasets** ***(ICLR 2020)***

Tool: Differentially Private Federated Generative Models

- Using generative models instead of data inspection
- Generative models + Federated learning (FL) + Differential privacy (DP)
- Algorithm => ‘*DP-FedAvg-GAN*’

Application of example model classes

- DP Federated RNNs for Generating Natural Language Data
- DP Federated GANs for Generating Image Data

Results & Key takeaways:

- **Result:** practical solution for robust FL data pipelining - train generative models using federated methods with differential privacy, and then using these to synthesize new data samples that can be used to debug the underlying data pipelines
- **Key takeaways**:
  - Experiments applying DP federated generative models to these workflows is a promising direction for future work;
  - Federated generative models to be useful and broadly applicable => require minimal tuning

Server-orchestrated training loop:

parameters: round participation fraction $q \in (0, 1]$, total number of users $N \in \mathbb{N}$, total number of rounds $T \in \mathbb{N}$, noise scale $z \in \mathbb{R}^+$, clip parameter $S \in \mathbb{R}^+$

Initialize generator θ_G^0, discriminator θ_D^0, privacy accountant $\mathcal{M}$

Set $\sigma = \frac{zS}{qN}$

for each round t from 0 to T **do**

$\quad \mathcal{C}^t \leftarrow$ (sample of qN distinct users)

$\quad$**for** each user $k \in \mathcal{C}^t$ **in parallel do**

$\quad\quad \Delta_k^{t+1} \leftarrow \text{UserDiscUpdate}(k, \theta_D^t, \theta_G^t)$

$$\Delta^{t+1} = \frac{1}{qN} \sum_{k \in \mathcal{C}^t} \Delta_k^{t+1}$$

$$\theta_D^{t+1} \leftarrow \theta_D^t + \Delta^{t+1} + \mathcal{N}(0, I\sigma^2)$$

$\quad \mathcal{M}.\texttt{accum\_priv\_spending}(z)$

$$\theta_G^{t+1} \leftarrow \text{GenUpdate}(\theta_D^{t+1}, \theta_G^t)$$

print $\mathcal{M}.\texttt{get\_privacy\_spent}()$

Algorithm 1: DP-FedAvg-GAN

Observation and Insights - Non-Malicious Failure in FL

- **Just as with adversarial attacks, systems factors and data constraints also exacerbate non-malicious failures present in Federated Learning**
- **Any federated learning system still needs to inspect raw data & preprocessed in to training data even if data pipelines in federated learning only exist within each client; Data restrictions in federated learning makes detection of pipeline failures significantly more challenging**
- **Even if the data on a client is not intentionally malicious, it may have non malicious issues such as noisy features**

Conclusion

- Security & Robustness of Federated Learning involve many aspect
  - **Defending Against Attacks and Failures:**
    - Non Malicious failures in preprocessing and training pipelines
    - Explicit attacks target at training and deployment pipelines
  - **Open questions** - other challenges in FL:
    - improve communication efficiency
    - reduce uplink communication cost

Thank you for listening:)