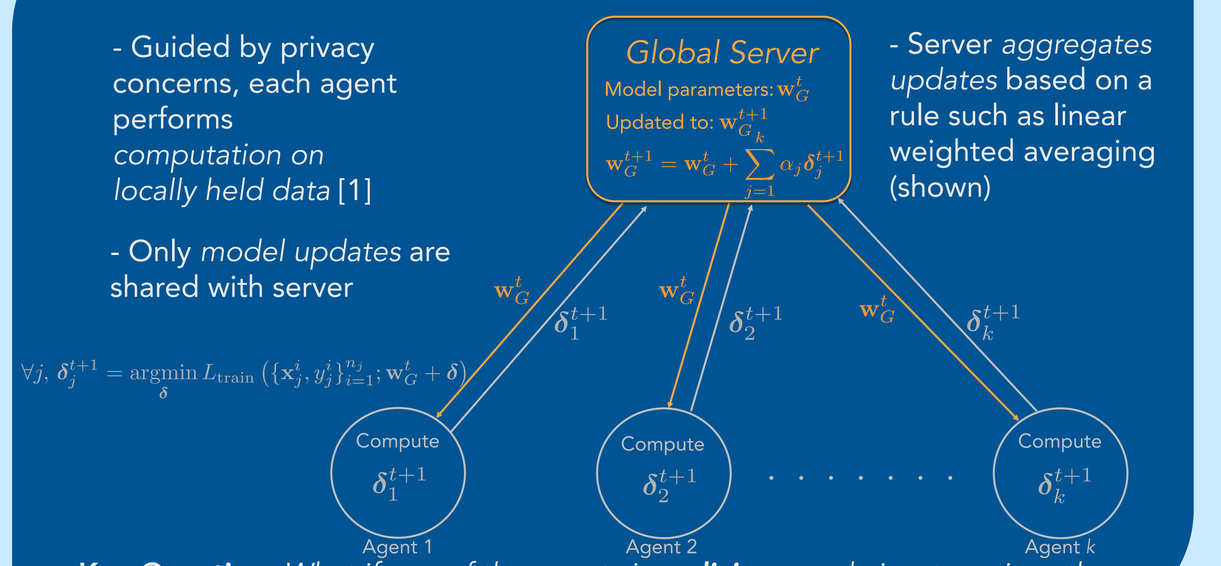
Analyzing Federated Learning through an Adversarial Lens

Arjun Nitin Bhagoji [1], Supriyo Chakraborty [2],
Prateek Mittal [1] and Seraphin Calo [2]
[1] Princeton University [2] I.B.M. Research

Federated Learning

- Guided by privacy concerns, each agent performs *computation on locally held data* [1]

- Only *model updates* are shared with server



- Server *aggregates updates* based on a rule such as linear weighted averaging (shown)

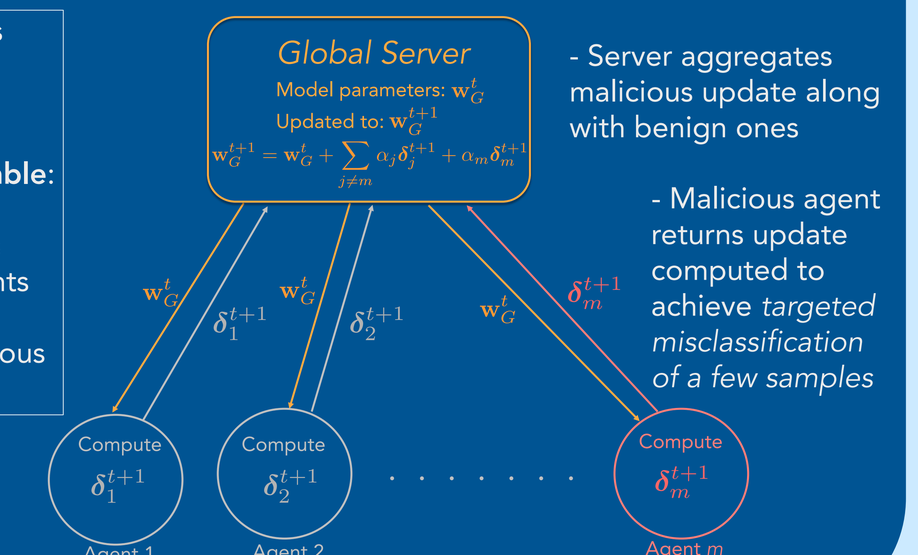
Key Question: *What if one of the agents is* ***malicious****, and aims to poison the learning process?*

Threat Model: Targeted Model Poisoning

- Single malicious agent: aims to poison global model

Information available:

- No access to current updates from other agents
- Attacks with respect to previous global state



- Server aggregates malicious update along with benign ones

- Malicious agent returns update computed to achieve *targeted misclassification of a few samples*

Attack Strategies for Model Poisoning

Eval. Setup

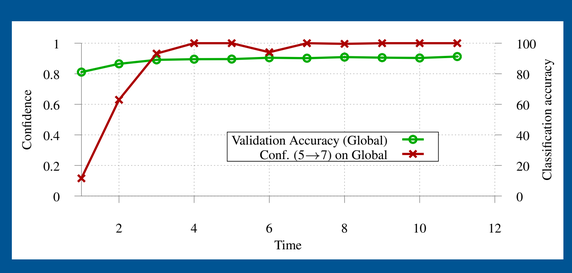
Dataset: Fashion MNIST [2]
Model: CNN with 91.7% test set accuracy

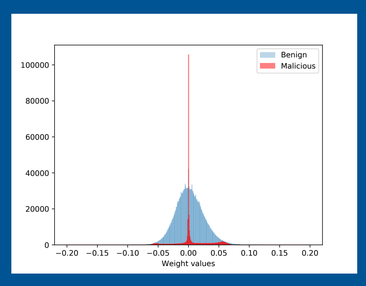
Malicious objective is to ensure  (sandal, class 5) is classified as a sneaker (class 7)

Targeted Model Poisoning

$$\boldsymbol{\delta}_{\text{mal}} = \underset{\delta}{\operatorname{argmin}} L_{\text{mal}} \left(\{\mathbf{x}^l, T^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta}\right)$$
$$\boldsymbol{\delta}_{\text{mal}} \rightarrow \beta \boldsymbol{\delta}_{\text{mal}}$$

- Compute update w.r.t. malicious objective
- Boost update when sending back to server



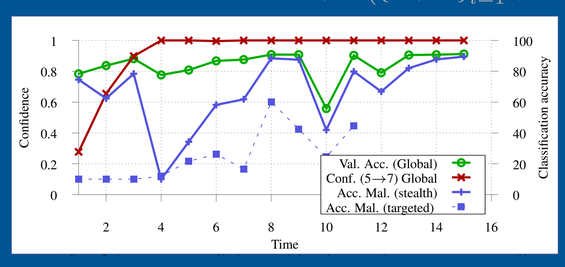


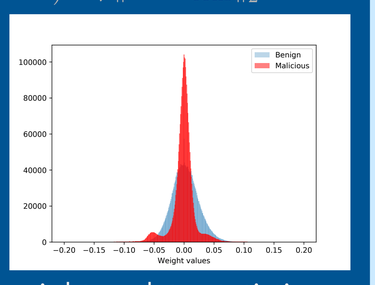
Takeaway: Malicious objective is met with high confidence while ensuring global model convergence **but** *malicious update clearly distinguishable*

Concatenated training

$$\boldsymbol{\delta}_{\text{mal}} = \underset{\delta}{\operatorname{argmin}} L\left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n_m}; \mathbf{w}_G + \boldsymbol{\delta}\right) + \beta L\left(\{\mathbf{x}^l, T^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta}\right) + \rho \|\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{cons}}\|_2^2$$

- Add benign training and distance constraints
- Boost only malicious component





Takeaway: Malicious agent is closer in accuracy and weight update statistics to benign agents **but** *convergence is erratic*

Alternating minimization with distance constraints

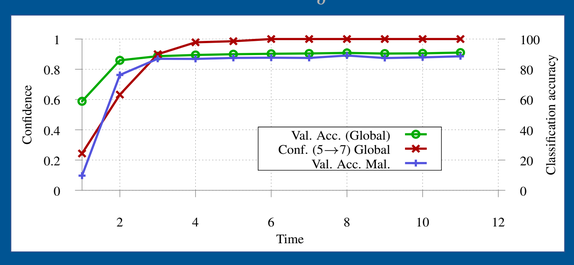
$$\boldsymbol{\delta}'_{\text{mal}} = \underset{\delta}{\operatorname{argmin}} L_{\text{mal}} \left(\{\mathbf{x}^l, T^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta}\right)$$

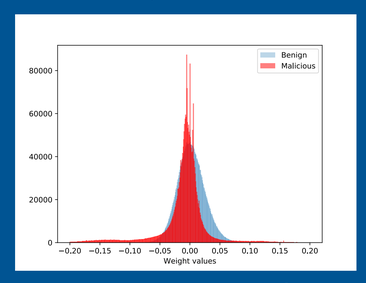
Repeat:

$$\boldsymbol{\delta}'_{\text{mal}} \rightarrow \beta \boldsymbol{\delta}'_{\text{mal}}$$

$$\boldsymbol{\delta}''_{\text{mal}} = \underset{\delta}{\operatorname{argmin}} L_{\text{ben}} \left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n_m}; \mathbf{w}_G + \beta \boldsymbol{\delta}'_{\text{mal}} + \boldsymbol{\delta}\right) + \rho \|\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{cons}}\|_2^2$$

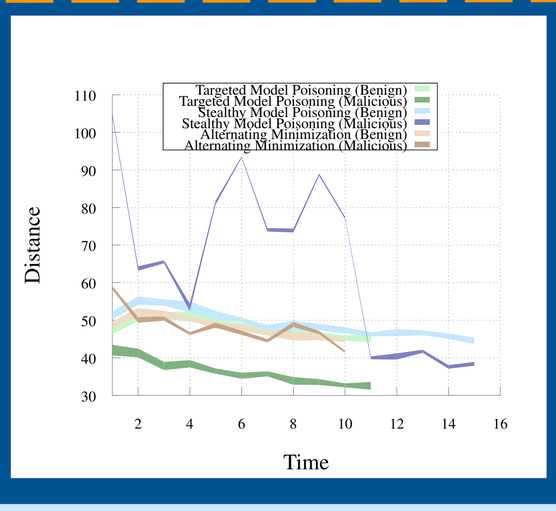
- Alternate between malicious and benign objectives
- Can control number of steps for each





Takeaway: Tighter control over the two objectives leads to targeted model poisoning with stealth in both accuracy and weight update statistics

Attack stealth measure: distance spread



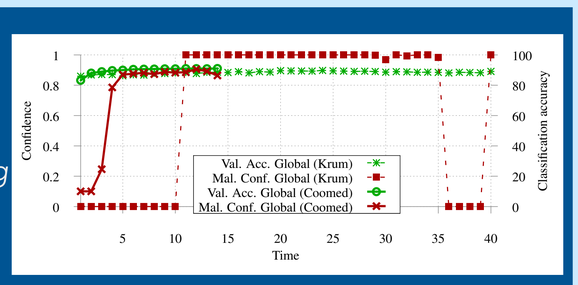
For each strategy, we show the spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents.

Takeaway: Spread of distances for malicious agent with alternating minimization is almost indistinguishable from that between benign agents'

Attacking Byzantine-resilient aggregation

- **Krum**: chooses set of k-2 updates closest to each other
- **Coomed**: performs coordinate-wise median

Attack works without boosting since no model averaging



Takeaway: Model poisoning is effective against Byzantine-resilient aggregation

Estimation to improve attacks

Pre-optimization correction with previous step estimate of benign agents' effects

$$\hat{\mathbf{w}}_G^t = \hat{\mathbf{w}}_G^{t-1} + \hat{\boldsymbol{\delta}}_{[k]\setminus m} + \alpha_m \boldsymbol{\delta}_m^t$$
$$\hat{\boldsymbol{\delta}}_{[k]\setminus m} = \boldsymbol{\delta}_{[k]\setminus m}^{t-1}$$

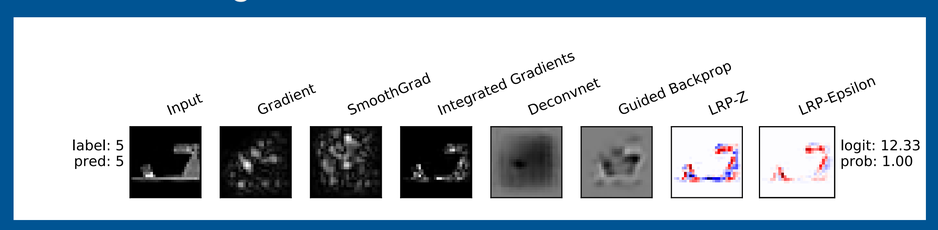
Attack	Targeted model poisoning		Alternating minimization	
Estimation	None	Prev. Step	None	Prev. Step
t=2	0.63	0.82	0.17	0.47
t=3	0.93	0.98	0.34	0.89
t=4	0.99	1.0	0.88	1.0

Takeaway: Estimation increases attack effectiveness, making it stronger earlier

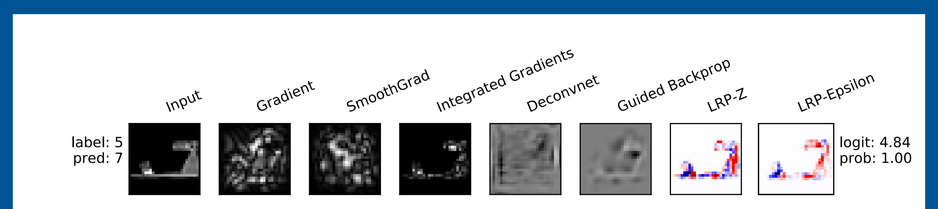
Interpreting Poisoned Models

- Interpretability techniques [3] provide insights into the internal feature representations and working of a neural network

Global model trained using only 10 benign agents



Global model trained with one malicious model among 10



Takeaway: Relevant input features used by the two models are almost visually imperceptible, further exposing the fragility of interpretability [4]

Conclusion

- Federated learning is very vulnerable to model poisoning attacks
- Detection mechanisms can make these attacks more challenging but these can be overcome
- **Open research question**: Can we develop distributed learning algorithms robust to model poisoning attacks?

References

[1] McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS 2017
[2] Xiao et al., *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*, arXiv preprint arXiv:1708.07747, 2017
[3] Alber et al., *iNNvestigate neural networks!*, arXiv preprint arXiv:1808.04260, 2018
[4] Adebayo et al., *Sanity checks for saliency maps*, NeurIPS 2018