FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping

Xiaoyu Cao, et al

Presented by Honglu Li

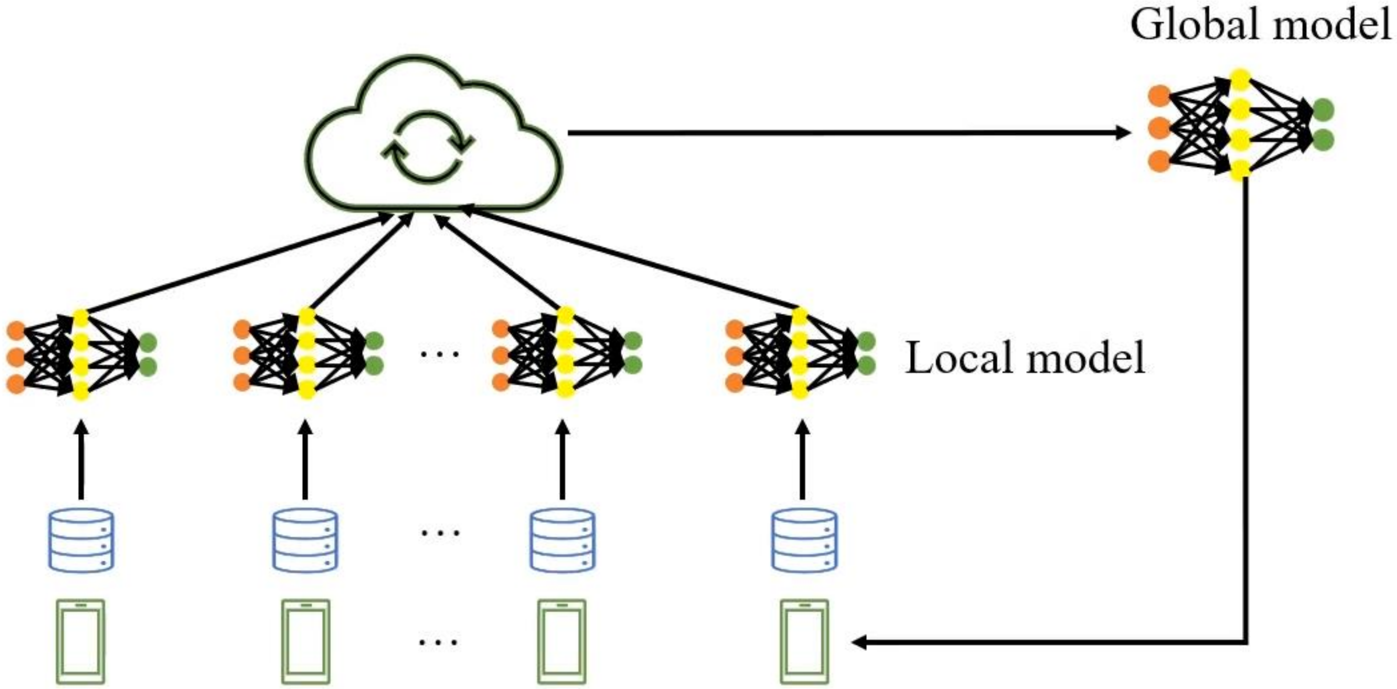
Published in ISOC Network and Distributed System Security Symposium (NDSS), 2021

Outline

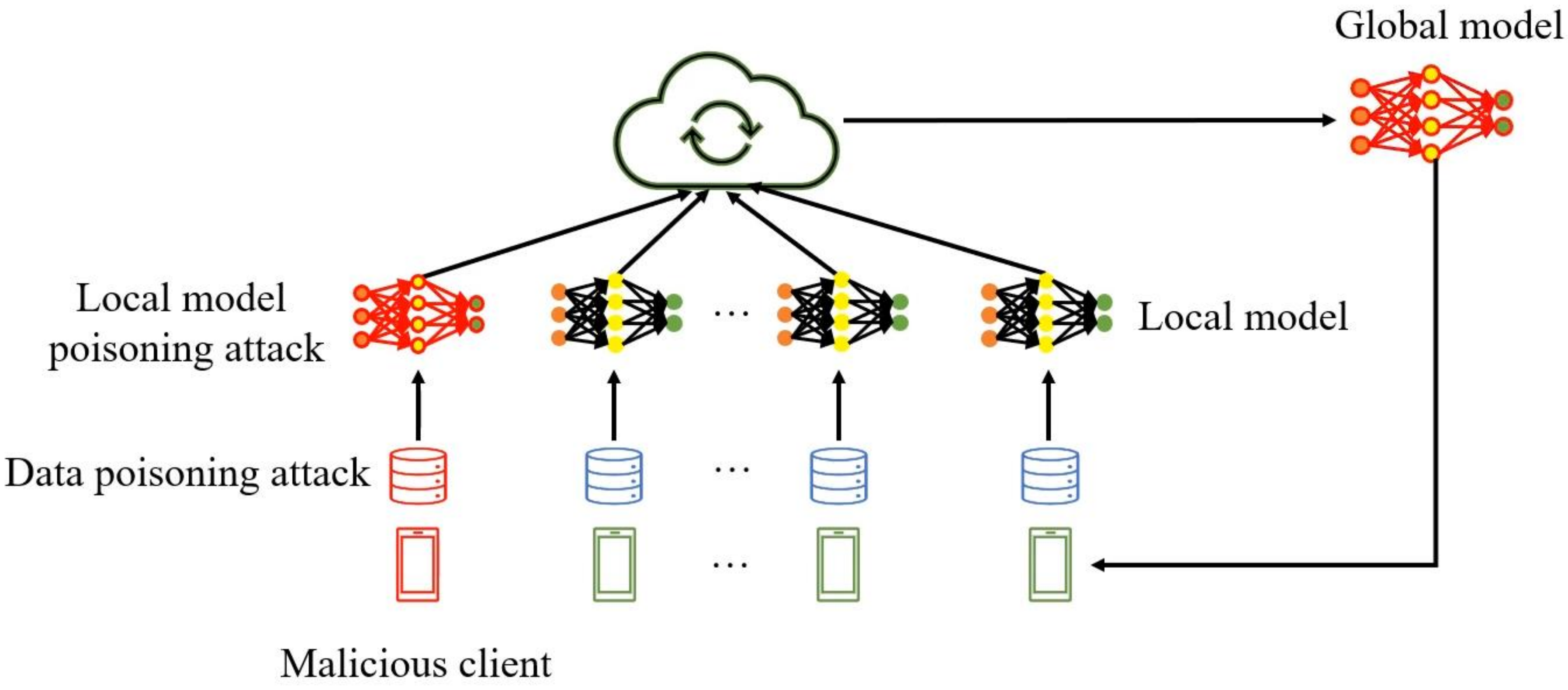
- Motivation
- FLTrust Design
- Evaluation
- Discussion

Motivation

IntelliSys Lab



Motivation


Global model
Local model
poisoning attack
Local model
Data poisoning attack
Malicious client

Motivation

- Byzantine-robust aggregation rule
  - Krum
  - Trimmed mean
  - Median
- Key idea
  - Remove “outlier” local model updates
- Byzantine-robust aggregation rule
  - Various assumptions
    - IID data, smooth loss function, etc.
  - Bound change of global model parameters caused by malicious clients

Existing Methods are Insecure

- Vulnerable to strong attacks
  - Local model poisoning attacks [1]
  - Backdoor attacks [2]

- Root cause
  - No root of trust
  - Every client could be malicious

[1] M. Fang, X. Cao, J. Jia, and N. Z. Gong, “Local model poisoning attacks to byzantine-robust federated learning,” in USENIX Security Symposium, 2020.
[2] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in AISTATS, 2020, pp. 2938–2948.

FLTrust: Booststrapping Trust

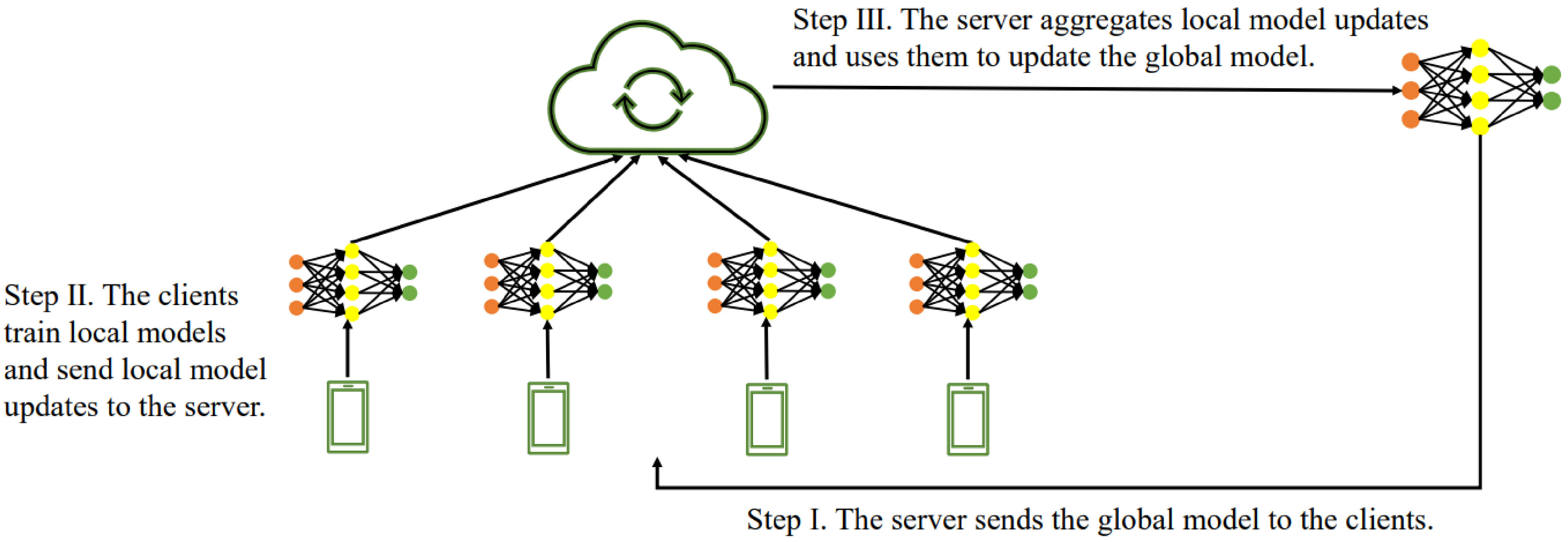
- Server collects a small clean training dataset
- Server maintains a server model
  - Like how a client maintains a local model
- Use server model update to bootstrap trust
  - Assign trust scores for clients

Outline

- Motivation
- FLTrust Design
- Evaluation
- Discussion

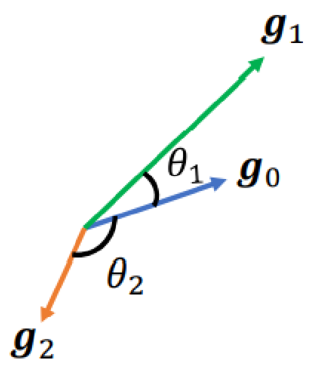
IntelliSys Lab

Three Steps in Federated Learning



New Aggregation Rule

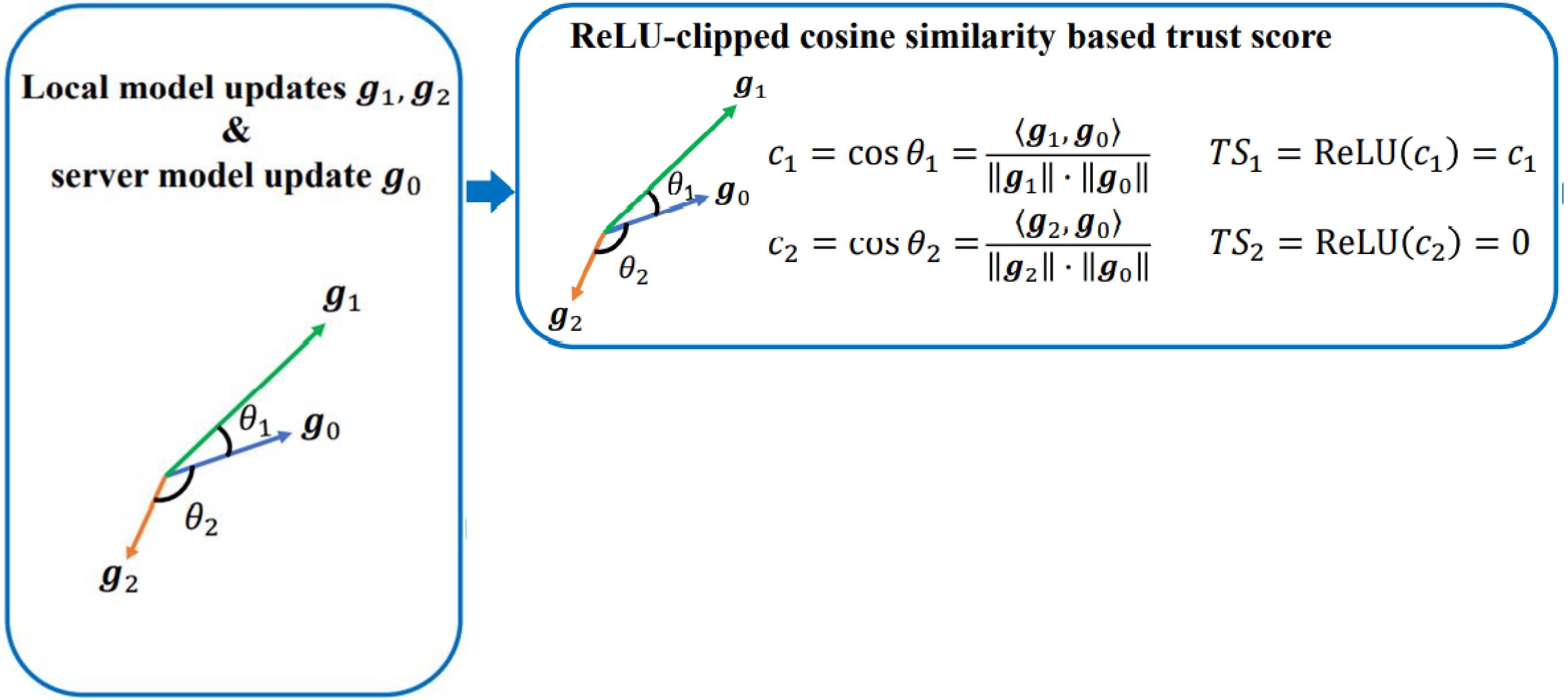
Local model updates $\boldsymbol{g}_1, \boldsymbol{g}_2$
&
server model update $\boldsymbol{g}_0$



Aggregation Rule

IntelliSys Lab

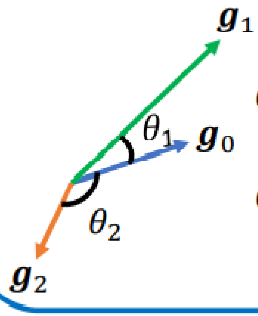
ReLU-clipped Cosine Similarity Based Trust Score



Aggregation Rule

ReLU-clipped Cosine Similarity Based Trust Score

ReLU-clipped cosine similarity based trust score



$$c_1 = \cos\theta_1 = \frac{\langle \boldsymbol{g}_1, \boldsymbol{g}_0 \rangle}{\|\boldsymbol{g}_1\| \cdot \|\boldsymbol{g}_0\|} \qquad TS_1 = \mathrm{ReLU}(c_1) = c_1$$

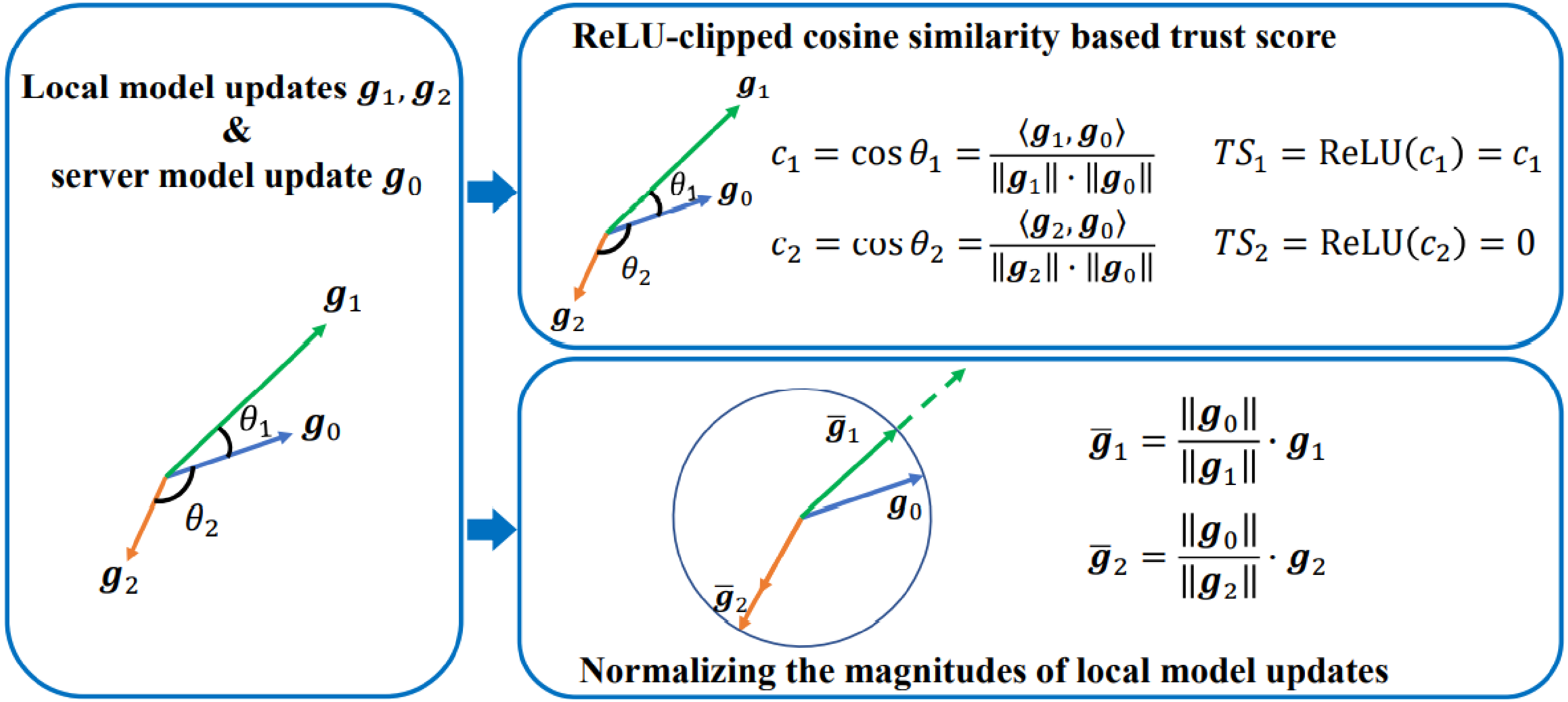
$$c_2 = \cos\theta_2 = \frac{\langle \boldsymbol{g}_2, \boldsymbol{g}_0 \rangle}{\|\boldsymbol{g}_2\| \cdot \|\boldsymbol{g}_0\|} \qquad TS_2 = \mathrm{ReLU}(c_2) = 0$$

$$TS_i = ReLU(c_i)$$

$$c_i = \frac{\langle \boldsymbol{g}_i, \boldsymbol{g}_0 \rangle}{||\boldsymbol{g}_i|| \cdot ||\boldsymbol{g}_0||}$$

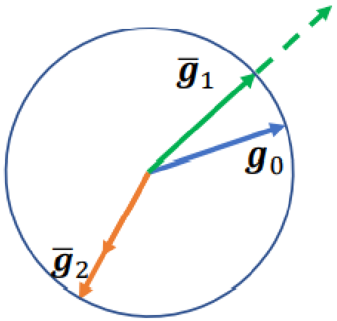
$$ReLU(x) = x \text{ if } x > 0$$
$$ReLU(x) = 0 \text{ otherwise}$$

Normalizing the Magnitudes of Local Model Updates



Aggregation Rule

Normalizing the Magnitudes of Local Model Updates



$$\bar{g}_1 = \frac{\|g_0\|}{\|g_1\|} \cdot g_1$$

$$\bar{g}_2 = \frac{\|g_0\|}{\|g_2\|} \cdot g_2$$

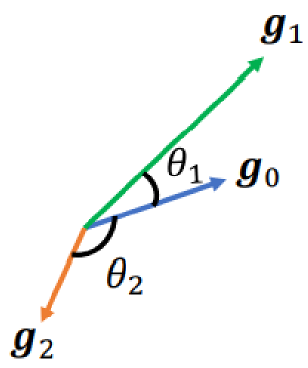
Normalizing the magnitudes of local model updates

$$\bar{g}_i = \frac{||g_0||}{||g_i||} \cdot g_i$$

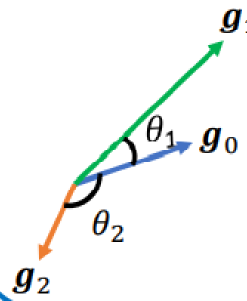
IntelliSys Lab

Aggregating the Local Model Updates

Local model updates $\boldsymbol{g}_1, \boldsymbol{g}_2$
&
server model update $\boldsymbol{g}_0$

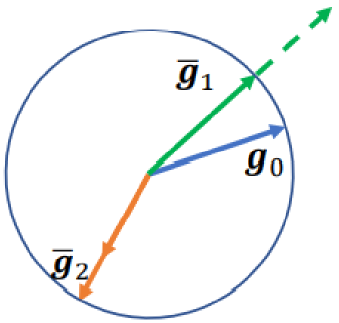


ReLU-clipped cosine similarity based trust score



$$c_1 = \cos\theta_1 = \frac{\langle \boldsymbol{g}_1, \boldsymbol{g}_0 \rangle}{\|\boldsymbol{g}_1\| \cdot \|\boldsymbol{g}_0\|} \qquad TS_1 = \mathrm{ReLU}(c_1) = c_1$$

$$c_2 = \cos\theta_2 = \frac{\langle \boldsymbol{g}_2, \boldsymbol{g}_0 \rangle}{\|\boldsymbol{g}_2\| \cdot \|\boldsymbol{g}_0\|} \qquad TS_2 = \mathrm{ReLU}(c_2) = 0$$



$$\bar{\boldsymbol{g}}_1 = \frac{\|\boldsymbol{g}_0\|}{\|\boldsymbol{g}_1\|} \cdot \boldsymbol{g}_1$$

$$\bar{\boldsymbol{g}}_2 = \frac{\|\boldsymbol{g}_0\|}{\|\boldsymbol{g}_2\|} \cdot \boldsymbol{g}_2$$

Normalizing the magnitudes of local model updates

Aggregation

$$\boldsymbol{g} = \frac{1}{TS_1 + TS_2}(TS_1 \cdot \bar{\boldsymbol{g}}_1 + TS_2 \cdot \bar{\boldsymbol{g}}_2)$$

Aggregating the Local Model Updates

Aggregation

$$\boldsymbol{g} = \frac{1}{TS_1 + TS_2}(TS_1 \cdot \bar{\boldsymbol{g}}_1 + TS_2 \cdot \bar{\boldsymbol{g}}_2)$$

$$\boldsymbol{g} = \frac{1}{\sum_{j=1}^{n} TS_j} \sum_{i=1}^{n} TS_i \cdot \bar{\boldsymbol{g}}_i$$

$$= \frac{1}{\sum_{j=1}^{n} ReLU(c_j)} \sum_{i=1}^{n} ReLU(c_i) \cdot \frac{||\boldsymbol{g}_0||}{||\boldsymbol{g}_i||} \cdot \boldsymbol{g}_i$$

Security Analysis

- Under some assumptions on learning problem
  - The expected loss function $F(w)$ is μ-strongly convex and differentiable over the space Θ with L-Lipschitz continuous gradient.
  - The gradient of the empirical loss function $\nabla f(D, w^*)$at the optimal global model w^* is bounded.
  - Each client's local training dataset D_i and the root dataset D_0 are sampled independently from the training data distribution.

- For an arbitrary number of malicious clients, the difference between the learnt global model and the optimal model under no attack is bounded.

Security Analysis

- Suppose the three assumptions hold and FLTrust uses R_l = 1 and β = 1, and let α be the combined learning rate.

- Lemma 1: For an arbitrary number of malicious clients, the distance between g and $\nabla F(w)$ is bounded as follows in each iteration:

$$\|\boldsymbol{g} - \nabla F(\boldsymbol{w})\| \leq 3\,\|\boldsymbol{g}_0 - \nabla F(\boldsymbol{w})\| + 2\,\|\nabla F(\boldsymbol{w})\| \quad (1)$$

Security Analysis

- Lemma 2: Assume Assumption 1 holds. If set the learning rate as $\alpha = \mu/(2L^2)$, then we have the following in any global iteration $t \geq 1$:

$$\left\| \boldsymbol{w}^{t-1} - \boldsymbol{w}^* - \alpha \nabla F(\boldsymbol{w}^{t-1}) \right\| \leq \sqrt{1 - \mu^2/(4L^2)} \left\| \boldsymbol{w}^{t-1} - \boldsymbol{w}^* \right\| \quad (2)$$

Security Analysis

- Lemma 3: Suppose Assumption 2 holds. For any $\delta \in (0,1)$ and any $w \in \Theta$, let

$$\Delta_1 = \sqrt{2}\sigma_1\sqrt{(d\log 6 + \log(3/\delta))/|D_0|}$$

$$\Delta_3 = \sqrt{2}\sigma_2\sqrt{(d\log 6 + \log(3/\delta))/|D_0|}$$

We have

$$Pr\left\{\left\|\frac{1}{|D_0|}\sum_{X_i \in D_0} \nabla f(X_i, \boldsymbol{w}^*) - \nabla F(\boldsymbol{w}^*)\right\| \geq 2\Delta_1\right\} \leq \frac{\delta}{3} \quad (3)$$

$$Pr\left\{\left\|\frac{1}{|D_0|}\sum_{X_i \in D_0} \nabla h(X_i, \boldsymbol{w}) - \mathbb{E}\left[h(X, \boldsymbol{w})\right]\right\| \geq 2\Delta_3 \|\boldsymbol{w} - \boldsymbol{w}^*\|\right\} \leq \frac{\delta}{3} \quad (4)$$

IntelliSys Lab

Security Analysis

- Lemma 4: Suppose Assumptions 1-3 hold, Then, for any $\delta \in (0,1)$, if $\Delta_1 \leq \sigma_1{}^2/\gamma_1$ and $\Delta_2 \leq \sigma_2{}^2/\gamma_2$, we have the following for any $w \in \Theta$:

$$Pr\{\|\boldsymbol{g}_0 - \nabla F(\boldsymbol{w})\| \leq 8\Delta_2\|\boldsymbol{w} - \boldsymbol{w}^*\| + 4\Delta_1\} \geq 1 - \delta \quad (5)$$

$$where\ \Delta_2 = \sigma_2\sqrt{\frac{2}{|D_0|}}\sqrt{K_1 + K_2},\ K_1 = d\log\frac{18L_2}{\sigma_2}$$

$$K_2 = \frac{1}{2}d\log\frac{|D_0|}{d} + \log\left(\frac{6\sigma_2^2 r\sqrt{|D_0|}}{\gamma_2\sigma_1\delta}\right),\ L_2 = \max\{L, L_1\}$$

Security Analysis

- With the lemmas above, we can prove the difference between the global model learnt by FLTrust and the optimal global model w^* under no attacks is bounded.

$$\left\|\boldsymbol{w}^t - \boldsymbol{w}^*\right\| \leq (1-\rho)^t \left\|\boldsymbol{w}^0 - \boldsymbol{w}^*\right\| + 12\alpha\Delta_1/\rho \qquad (6)$$

Adaptive Attacks

- Local model poisoning attacks [1]

$$\max_{\boldsymbol{w}'_1,\ldots,\boldsymbol{w}'_c} \boldsymbol{s}^T(\boldsymbol{w}-\boldsymbol{w}')$$

Subject to $\quad \boldsymbol{w} = \mathcal{A}(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c, \boldsymbol{w}_{c+1}, \ldots, \boldsymbol{w}_n)$

$$\boldsymbol{w}' = \mathcal{A}(\boldsymbol{w}'_1, \ldots, \boldsymbol{w}'_c, \boldsymbol{w}_{c+1}, \ldots, \boldsymbol{w}_n)$$

[1] M. Fang, X. Cao, J. Jia, and N. Z. Gong, “Local model poisoning attacks to byzantine-robust federated learning,” in USENIX Security Symposium, 2020.

Outline

- Motivation
- FLTrust Design
- Evaluation
- Discussion

Experimental Setup

- Datasets
  - MNIST-0.1, MNIST-0.5, Fashion-MNIST, CIFAR-10, Human activity recognition (HAR) and CH-MNIST
- Poisoning attacks
  - Label flipping (LF) attack, Krum attack, Trim attack, Scaling attack and Adaptive attack
- Global models
  - CNN, LR, ResNet20

Parameter Settings

	Explanation	MNIST-0.1	MNIST-0.5	Fashion-MNIST	CIFAR-10	HAR	CH-MNIST
n	# clients	100				30	40
τ	# clients selected in each iteration	n					
R_l	# local iterations	1					
R_g	# global iterations	2,000		2,500	1,500	1,000	2,000
b	batch size	32			64	32	
$\alpha \cdot \beta$	combined learning rate	3×10^{-4}		6×10^{-3}	2×10^{-4}	3×10^{-3}	3×10^{-4} (decay at the 1500th and 1750th iterations with factor 0.9)
m/n	fraction of malicious clients (%)	20					
m	# malicious clients	20				6	8
f	Krum parameter	m					
k	Trim-mean parameter	m					
$\lvert D_0 \rvert$	size of the root dataset	100					

Evaluation

Different Federated Learning Methods

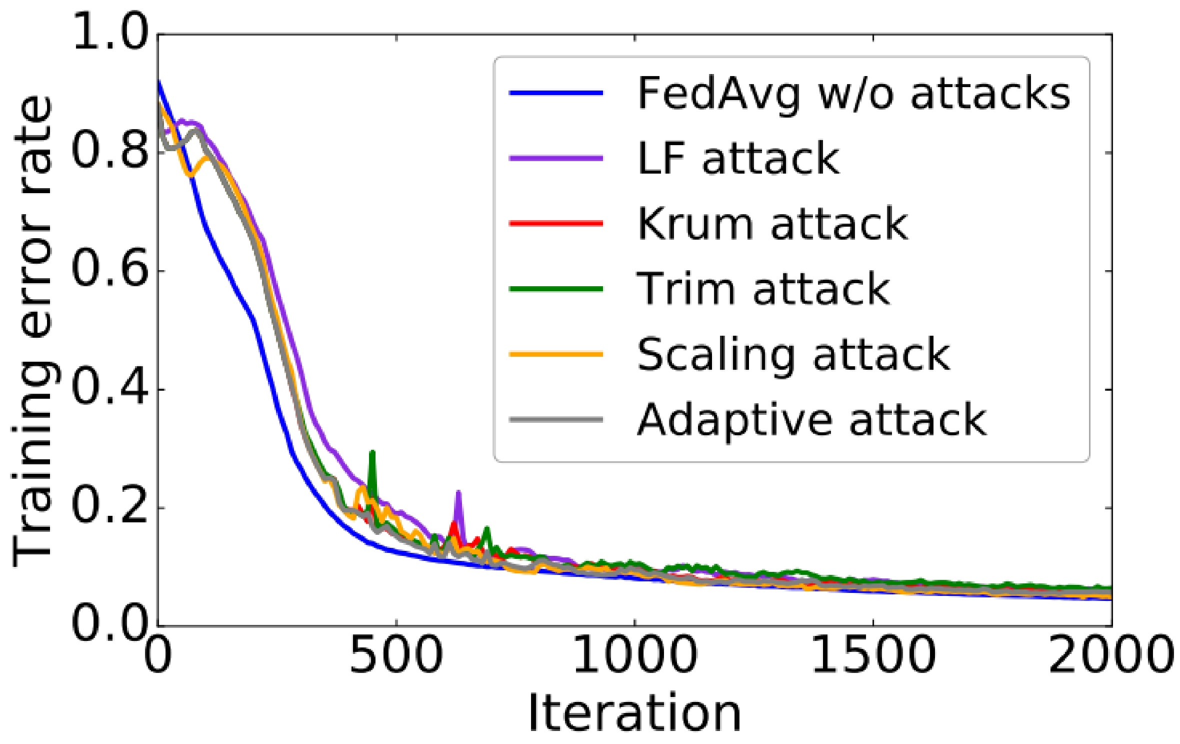
	FedAvg	Krum	Trim-mean	Median	FLTrust
No attack	0.04	0.10	0.06	0.06	0.05
LF attack	0.06	0.10	0.06	0.06	0.05
Krum attack	0.10	0.91	0.14	0.15	0.05
Trim attack	0.28	0.10	0.23	0.43	0.06
Scaling attack	0.02 / 1.00	0.09 / 0.01	0.06 / 0.02	0.06 / 0.01	0.05 / 0.00
Adaptive attack	0.13	0.10	0.22	0.90	0.06

- MNIST
- 100 clients, 20 malicious
- Root dataset: 100 training examples

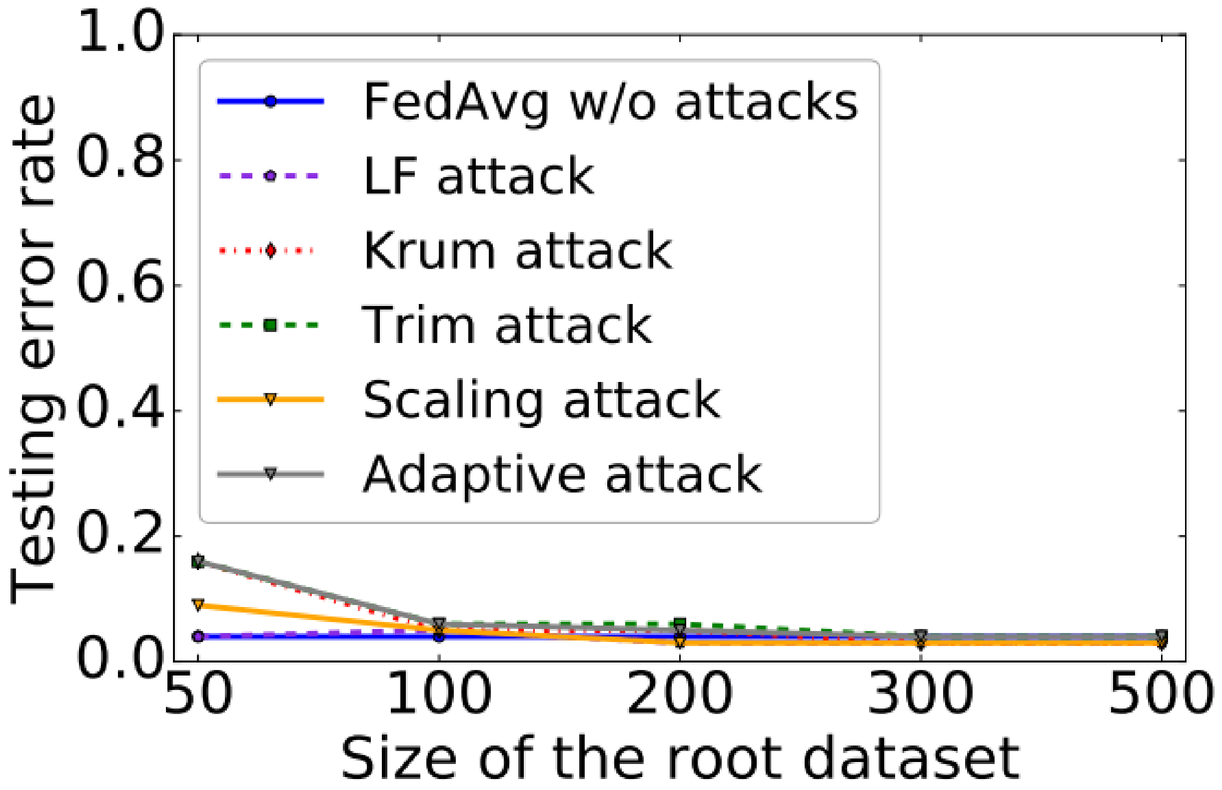
Five Variants of FLTrust

	No attack	LF attack	Krum attack	Trim attack	Scaling attack	Adaptive attack
FLTrust-Server	0.21	–	–	–	–	–
FLTrust-withServer	0.07	0.08	0.09	0.10	0.08 / 0.01	0.94
FLTrust-NoReLU	0.28	0.90	0.90	0.90	0.94 / 0.08	0.90
FLTrust-NoNorm	0.05	0.06	0.06	0.08	0.94 / 0.08	0.06
FLTrust-ParNorm	0.06	0.06	0.06	0.06	0.06 / 0.01	0.06
FLTrust	0.05	0.05	0.05	0.06	0.05 / 0.00	0.06

Number of Iterations

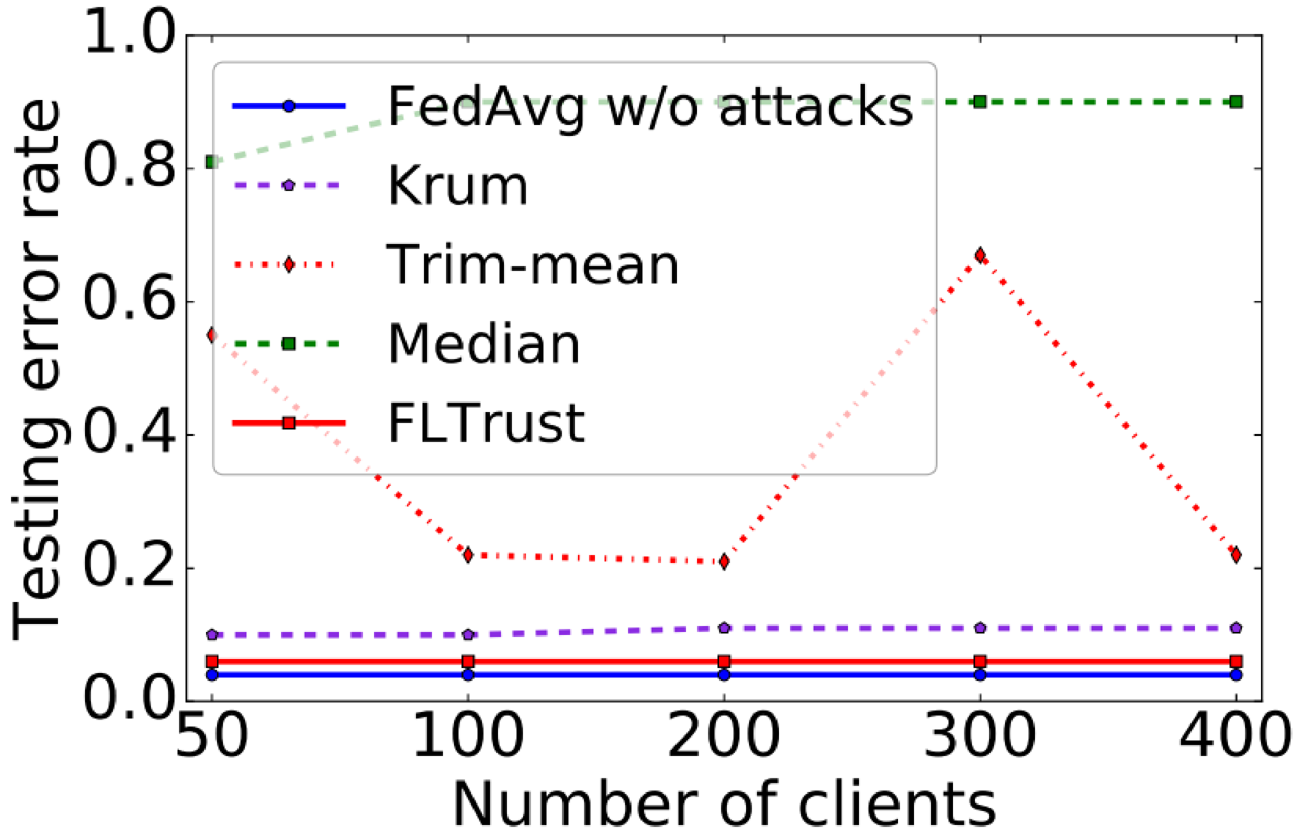


Root Dataset Size

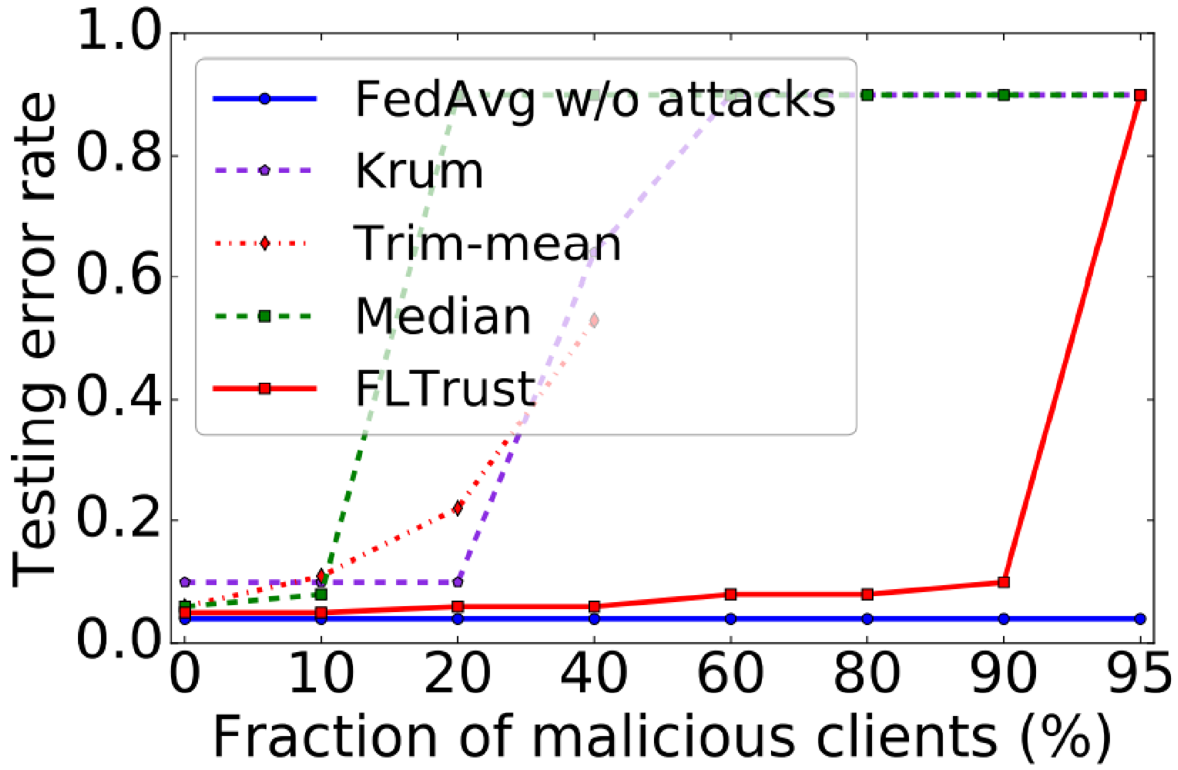


Evaluation

Number of Clients



Fraction of Malicious Clients



Outline

IntelliSys Lab

- Motivation
- FLTrust Design
- Evaluation
- Discussion

Discussion

- Poisoned root dataset
  - FLTrust requires a clean root dataset
  - FLTrust may not be robust against poisoned root dataset
- Adaptive attacks and hierarchical root of trust
  - There may exist stronger local model poisoning attacks to FLTrust, which is an interesting future work to explore
  - It is an interesting future work to consider a hierarchical root of trust

Conclusion

- This paper proposed and evaluated a new federated learning method called FLTrust to achieve Byzantine robustness against malicious clients

- Evaluations on six datasets show that FLTrust with a small root dataset can achieve Byzantine robustness against a large fraction of malicious clients

Thank You