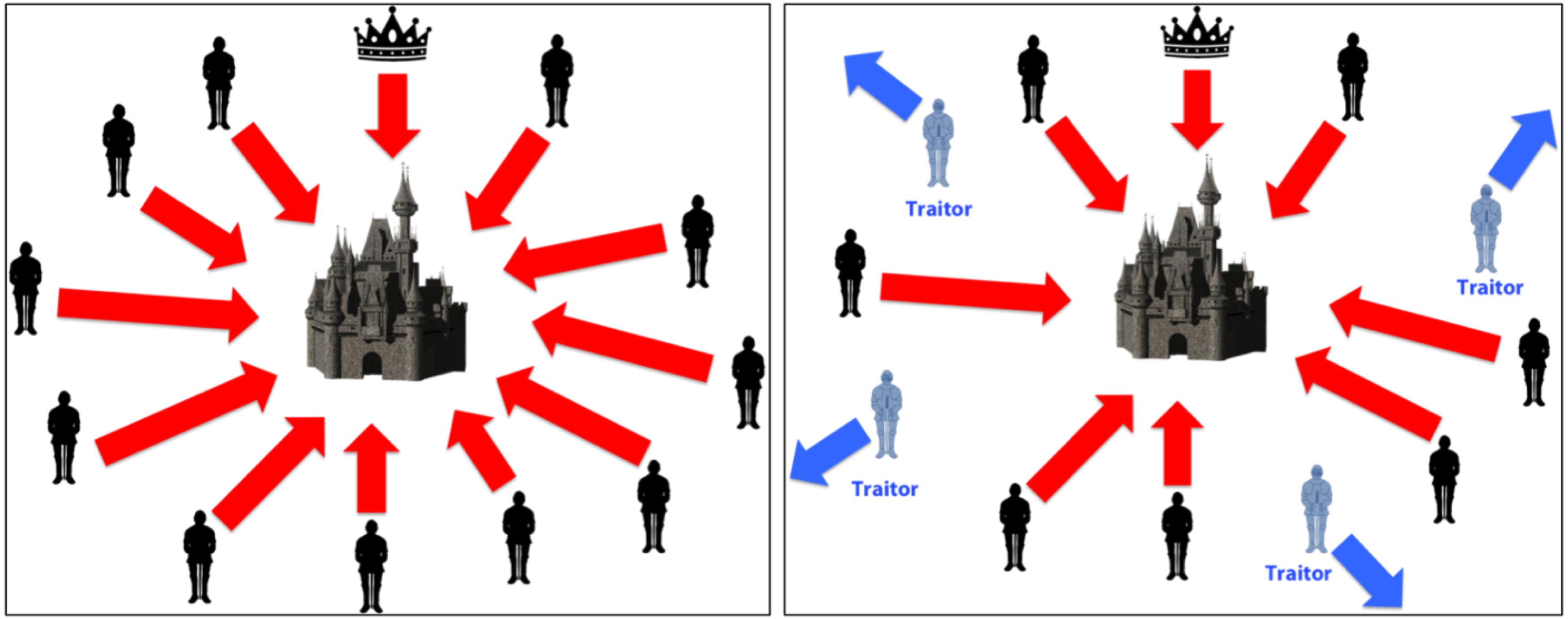
Local Model Poisoning Attacks to Byzantine-Robust Federated Learning

Feb 13, 2023

Jian, Pingcheng; Zhang, Yixin

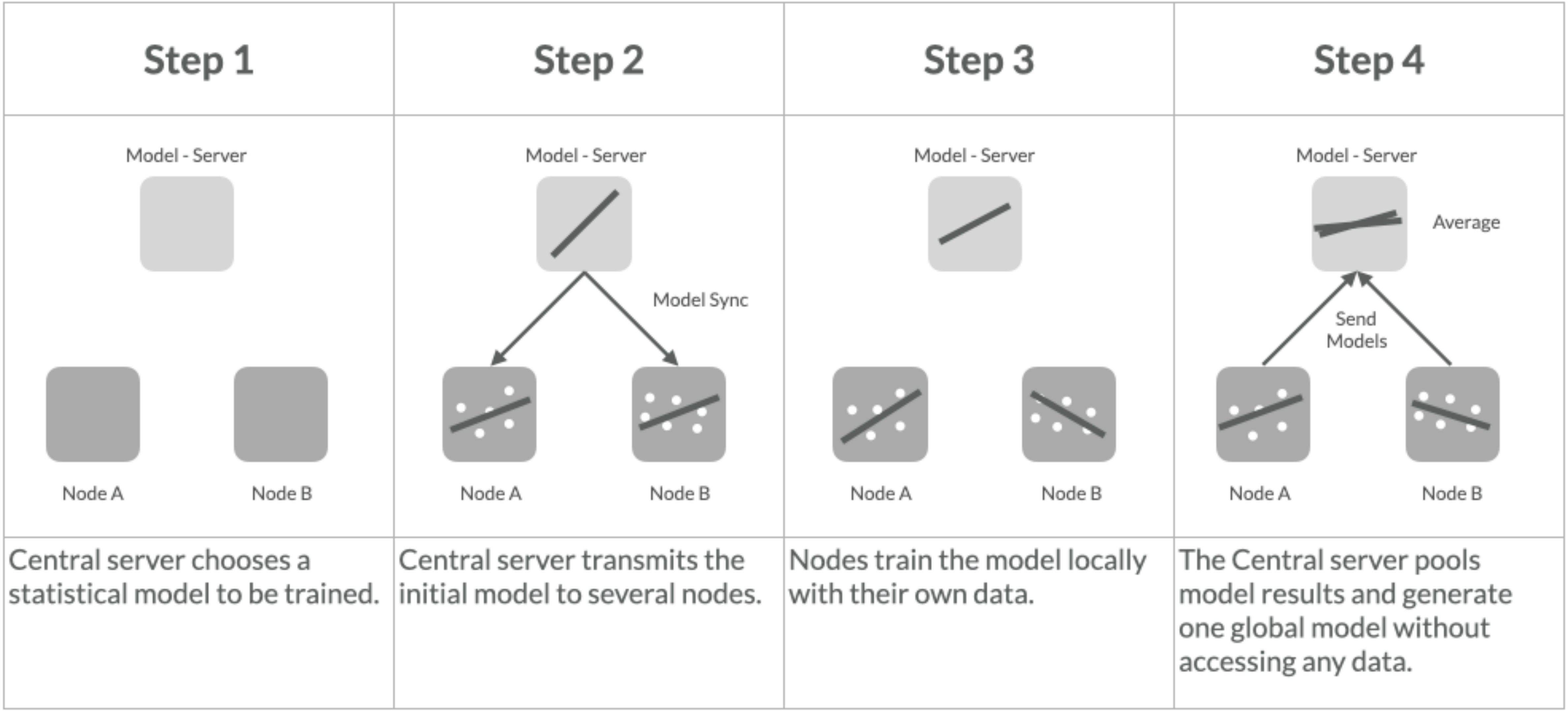
Byzantine failure (Byzantine Generals Problem)

Several generals are besieging Byzantium. They have surrounded the city, but they must collectively decide when to attack. If all generals attack at the same time, they will win, but if they attack at different times, they will lose.



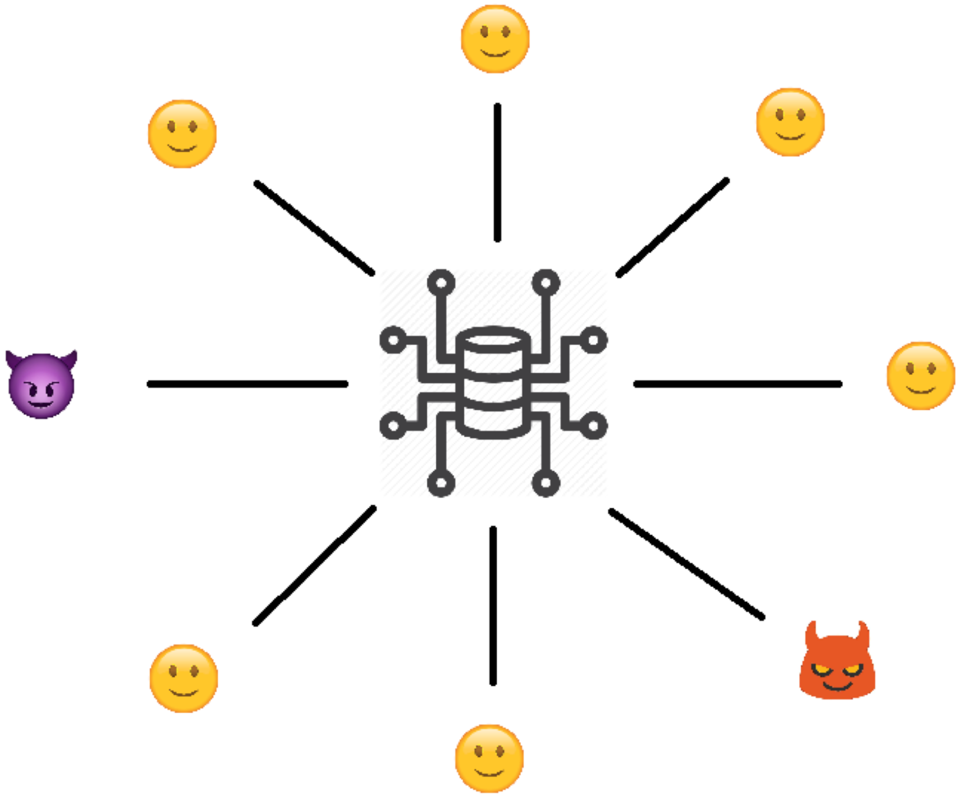
Coordinated Attack Leading to Victory **Uncoordinated Attack Leading to Defeat**

Federated Learning



Step 1	Step 2	Step 3	Step 4
Central server chooses a statistical model to be trained.	Central server transmits the initial model to several nodes.	Nodes train the model locally with their own data.	The Central server pools model results and generate one global model without accessing any data.

Byzantine failure in naïve Federated Learning



Mean aggregation is a typical method in non-adversarial training

Random Error is tolerable

(Paper) Label Noise in Segmentation Networks: Mitigation Must Deal with Bias

Attacker can use bad weight to easily counter the good weights from the benign parties

(i.e. $-w^{(t)} * l$, where l is the number of benign parties)

Threat model in Federated Learning

Attacker’s knowledge:

- Model architecture
- Local training set of compromised devices
- Local model weight of compromised devices
- parameter of global model

Attacker maybe know:

- Aggregation policy

Attacker do not know:

- Local training data on benign device
- Local model weight of benign device

Byzantine-robustness through aggregation

Predate this paper, the prevalent defense algorithms for federated learning were through Byzantine-robust Aggregation.

- Krum and Bulyan (aggregation by election)
- Trimmed Mean
- Median

We will first introduce the rational behind the Byzantine-robust aggregation, and then discuss the attack models and attack implementations of these three (four) byzantine-robust algorithms.

Krum aggregation

Rule: pick one of the m local models that is similar to other models as the global model.

Intuition: If the picked model is benign, we are good. If the picked model is malicious, it is still “close” to a benign model.

Implementation:

Step 1: set a hyperparameter ‘c’ as the maximum # attacker

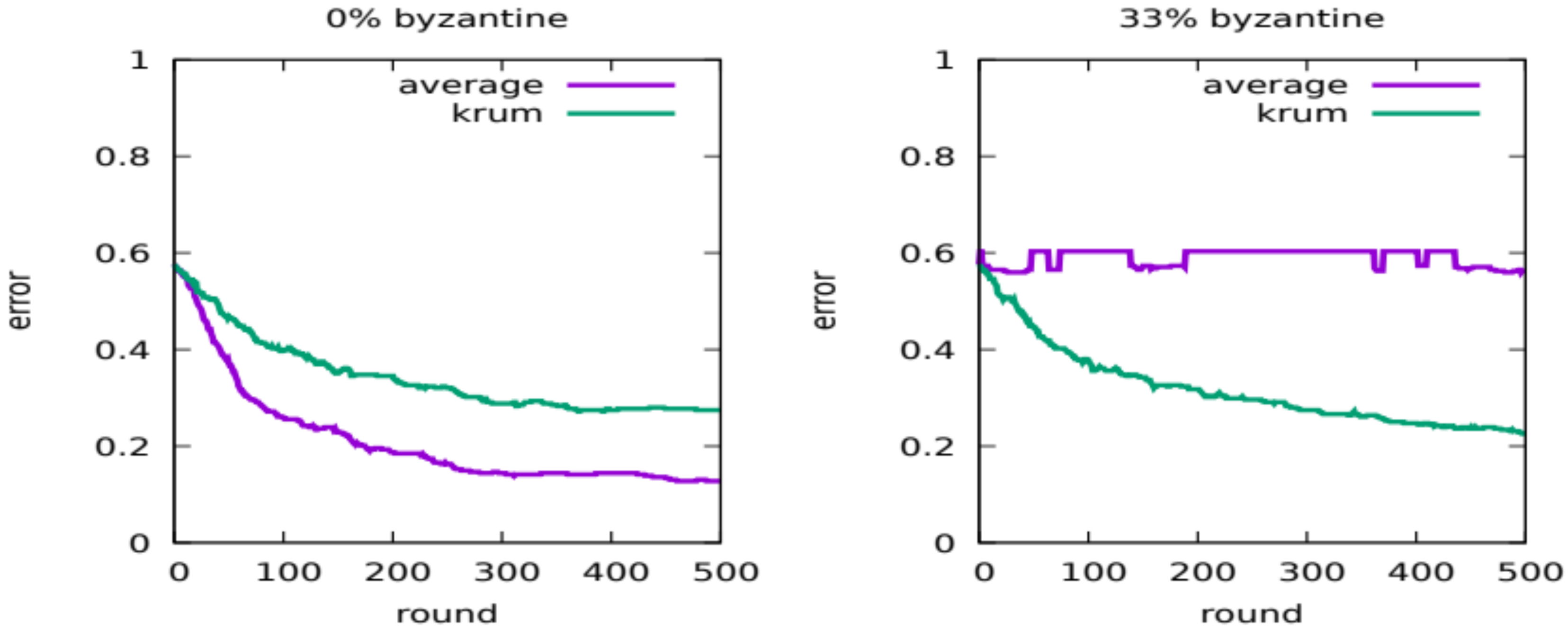
Step 2: for each model w_i, compute

$$D_i^{(t)} = \text{sum}\left(sort \left\{ \left\| w_i^{(t)} - w_l^{(t)} \right\|_2 \ \forall l \neq i \right\} [0{:}\, m - c - 2] \right)$$

Step 3: set $w^{(t+1)} = w_i^{(t)}$ with $i = argmin\left(D_1^{(t)}, \ldots, D_m^{(t)} \right)$

More notions about Krum

- Convergence to small gradient norm is guaranteed (converge to flat region)
- Though convergence of loss function is guaranteed by Krum, the convergence point is sub-optimal



Bulyan Aggregation

Intuition: L2 distance used in Krum is sensitive to outlier, so does extreme values in parameters

- To address this, Bulyan extends Krum with trimmed mean

Step 1:

select k models with lowest $D^{(t)}$

Step 2:

set the value for final model's parameter as the trimmed mean (the mean of n values closest to the median) for each parameter.

Trimmed Mean Aggregation

- Aggregates each model parameter independently
- Master device sorts the j^{th} parameters of the m local models, removes the largest and smallest β of them, and computes the mean of the remaining m−2β parameters as the j^{th} parameter of the global model
- achieves order-optimal error rate when $c \leq \beta \leq \frac{m}{2}$ and the objective function to be minimized is strongly convex
- the order-optimal error rate is $\tilde{O}(\frac{c}{m\sqrt{n}} + \frac{1}{\sqrt{mn}})$, where n is the number of training data points on a worker device

Median Aggregation

- For each j^{th} model parameter, the master device sorts the j^{th} parameters of the m local models and takes the median (mean of the middle two parameters when m is even) as the j^{th} parameter of the global model.
- Median aggregation rule also achieves an order-optimal error rate when the objective function is strongly convex.

Attack as an optimization problem

- Directed deviation Goal

$$\max \quad \boldsymbol{s}^T(\boldsymbol{w} - \boldsymbol{w}')$$

$\boldsymbol{s}$ represents the gradient direction in this iteration suppose no attack.

- Undirected deviation Goal (Appendix B)

$$\max \quad \left|\left|\boldsymbol{w} - \boldsymbol{w}'\right|\right|_1$$

Here,

$$\boldsymbol{w} = \mathcal{A}(w_1, \dots, w_c, w_{c+1}, \dots, w_m)$$
$$\boldsymbol{w}' = \mathcal{A}(w_1{}', \dots, w_c{}', w_{c+1}, \dots, w_m)$$

with (.)’ means adversarial weight and $\mathcal{A}$ means aggregation function.

Attack on Krum (not yet)

A random question:

What is the easiest way to make people think you are popular when you walk into a party that you know nobody?

Step 1:

Get to know a few people and be close to them

Step 2:

Bring in your friend -- a lot but less than half in the room.

Step 3:

Now, half of the people knows you – and the rest will think you are popular.

Attack on Krum (Full knowledge)

- The key to the attack on Krum is an analogy of the boring plot.
- Attack to Krum takes the form

$$w_1{}' = w_{Re} - \lambda \boldsymbol{s}$$

Effectively, the authors proposed a scaled fast-gradient-sign attack so that the adversarial model weight can be against the benign gradient direction.

$$0 \leq \lambda \leq \sqrt{\frac{\min\limits_{c+1 \leq i \leq m} \sum D^2(w_l, w_i)}{(m-2c-1)d}} + \cdots$$

If a suitable λ exists that will make $w_1{}'$ be accepted as the global model (appendix C) , then it can be found with binary search.

Attack on Krum (Partial knowledge)

- Surrogate s with $\tilde{s}$, which can be obtained by using the benign weight that should have been obtained from the devices controlled by the attacker.

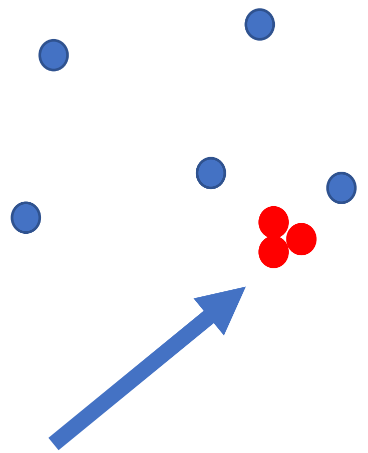
ERROR: MNIST	No Attack	Partial Knowledge	Full Knowledge
Logistic Regression	0.14	0.72	0.80
DNN	0.11	0.75	0.77

ERROR:F-MNIST	No Attack	Partial Knowledge	Full Knowledge
Logistic Regression	0.16	0.90	0.91
DNN	0.29	0.73	0.81

ERROR Rate	No Attack	Partial Knowledge	Full Knowledge
DNN: Cancer	0.29	0.73	0.81
DNN: CH-MNIST	0.03	0.17	0.17

Bulyan: Transferability from Krum

- The adversarial models distribute in the same epsilon ball with one of them as the center. The selection of the k models with the minimum Ds will mostly still be adversarial models. Hence, the next trimmed mean step is effective evaded.



Adversarial examples help each other to get small Ds

Attack on Trimmed Mean

General Idea:

- crafts the compromised local models based on the maximum or minimum benign local model parameters, depending on which one deviates the global model towards the inverse of the direction along which the global model would change without attacks.

With full knowledge :

- we calculate the maximum $w_{max,j}$ and minimum $w_{min,j}$ of the j^{th} local model parameters on the benign worker devices
- If the changing direction $s_j = -1$, we randomly sample the c numbers in the interval $[w_{max,j}, \mathrm{b} \cdot w_{max,j}]$ (when $w_{max,j} > 0$) or $[w_{max,j}, w_{max,j}/b]$ (when $w_{max,j} \leq 0$)
- If the $s_j = 1$, we randomly sample the c numbers in the interval $[w_{min,j}/b, w_{min,j}]$ (when $w_{min,j} > 0$) or $[\mathrm{b} \cdot w_{min,j}, w_{min,j}]$ (when $w_{min,j} \leq 0$).
- The sampled c numbers should be close to $w_{max,j}$ or $w_{min,j}$ to avoid being outliers and being detected easily.
- b=2. Attack does not depend on b once b > 1

Attack on Trimmed Mean

With partial knowledge :

- Problems: attacker does not know the changing direction variable s_j; attacker does not know the maximum $w_{max,j}$ and minimum $w_{min,j}$ of the j^{th} local model parameters on the benign worker devices
- estimate the changing direction s_j using the local models on the compromised worker devices
- estimate $w_{max,j}$ and $w_{min,j}$ using the before attack local model parameters on the compromised worker devices by computing the mean μ_j and standard deviation σ_j of each j^{th} parameter on the compromised worker devices
- estimate that $w_{max,j}$ is smaller than μ_j+3σ_j or μ_j+4σ_j with large probabilities; and $w_{min,j}$ is larger than μ_j−4σ_j or μ_j−3σ_j with large probabilities
- When s_j is estimated to be −1, we sample c numbers from the interval [μ_j+3σ_j, μ_j+4σ_j] as the j^{th} parameter of the c compromised local models
- When s_j is estimated to be 1, we sample c numbers from the interval [μ_j-3σ_j, μ_j-4σ_j]

Attack on Median

General Idea:

- Use the same attacks for trimmed mean to attack the median aggregation rule

Example for fully knowledge scenario:

- If the changing direction $s_j = -1$, we randomly sample the c numbers in the interval $[w_{max,j}, \mathrm{b} \cdot w_{max,j}]$ (when $w_{max,j} > 0$) or $[w_{max,j}, w_{max,j}/b]$ (when $w_{max,j} \leq 0$)
- If the $s_j = 1$, we randomly sample the c numbers in the interval $[w_{min,j}/b, w_{min,j}]$ (when $w_{min,j} > 0$) or $[\mathrm{b} \cdot w_{min,j}, w_{min,j}]$ (when $w_{min,j} \leq 0$).

Testing error rates of various attacks

- our attacks are effective and substantially outperform existing attacks

- Krum is less robust to our attacks than trimmed mean and median, except on Breast Cancer Wisconsin (Diagnostic) where Krum is comparable to median

- The error rates may depend on the data dimension: MNIST and Fashion-MNIST have 784 dimensions, CH-MNIST has 4096 dimensions, and Breast Cancer Wisconsin (Diagnostic) has 30 dimensions. For the DNN classifiers, the error rates are higher on CH-MNIST than on other datasets in most cases, while the error rates are lower on Breast Cancer Wisconsin (Diagnostic) than on other datasets in most cases.

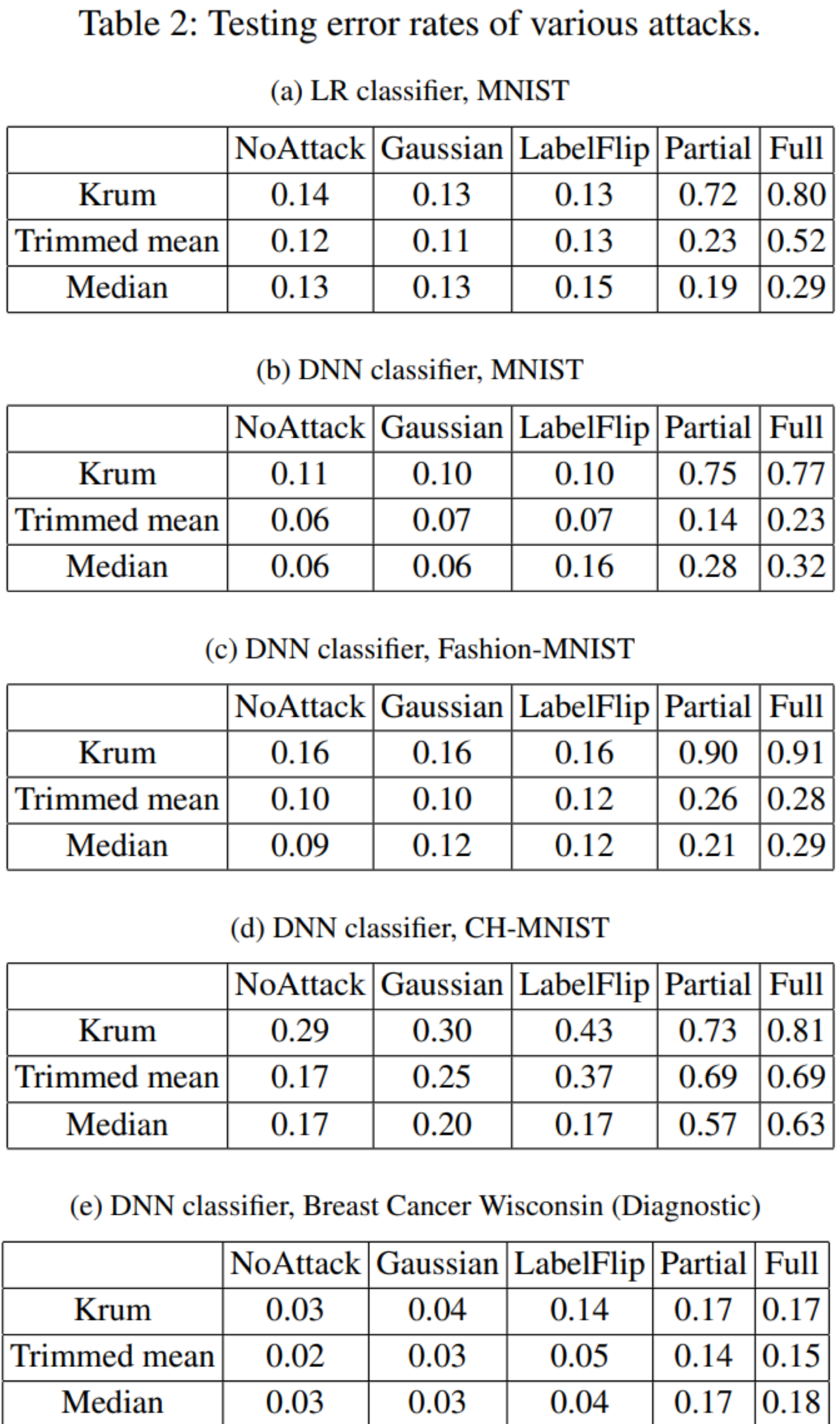


Table 2: Testing error rates of various attacks.

(a) LR classifier, MNIST

	NoAttack	Gaussian	LabelFlip	Partial	Full
Krum	0.14	0.13	0.13	0.72	0.80
Trimmed mean	0.12	0.11	0.13	0.23	0.52
Median	0.13	0.13	0.15	0.19	0.29

(b) DNN classifier, MNIST

	NoAttack	Gaussian	LabelFlip	Partial	Full
Krum	0.11	0.10	0.10	0.75	0.77
Trimmed mean	0.06	0.07	0.07	0.14	0.23
Median	0.06	0.06	0.16	0.28	0.32

(c) DNN classifier, Fashion-MNIST

	NoAttack	Gaussian	LabelFlip	Partial	Full
Krum	0.16	0.16	0.16	0.90	0.91
Trimmed mean	0.10	0.10	0.12	0.26	0.28
Median	0.09	0.12	0.12	0.21	0.29

(d) DNN classifier, CH-MNIST

	NoAttack	Gaussian	LabelFlip	Partial	Full
Krum	0.29	0.30	0.43	0.73	0.81
Trimmed mean	0.17	0.25	0.37	0.69	0.69
Median	0.17	0.20	0.17	0.57	0.63

(e) DNN classifier, Breast Cancer Wisconsin (Diagnostic)

	NoAttack	Gaussian	LabelFlip	Partial	Full
Krum	0.03	0.04	0.14	0.17	0.17
Trimmed mean	0.02	0.03	0.05	0.14	0.15
Median	0.03	0.03	0.04	0.17	0.18

transferability between aggregation rules

- MNIST and LR classifier are considered
- Krum based attack can well transfer to trimmed mean and median, e.g., Krum based attack increases the error rate from 0.12 to 0.15 (25% relative increase) for trimmed mean, and from 0.13 to 0.18 (38% relative increase) for median.
- Trimmed mean based attack does not transfer to Krum but transfers to median well. For instance, trimmed mean based attack increases the error rates from 0.13 to 0.20 (54% relative increase) for median.

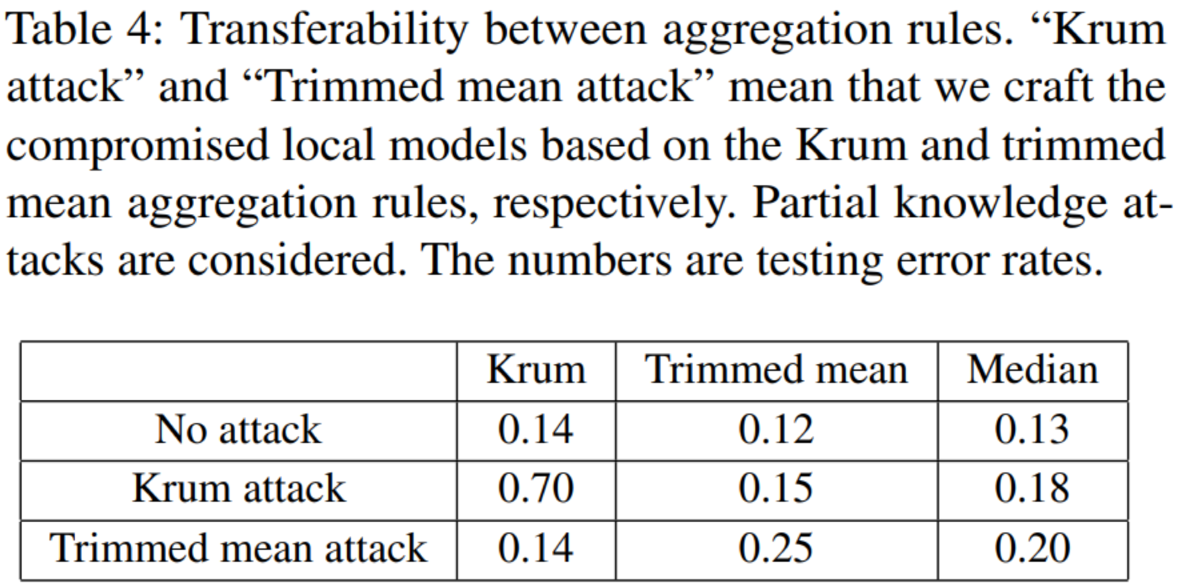


Table 4: Transferability between aggregation rules. "Krum attack" and "Trimmed mean attack" mean that we craft the compromised local models based on the Krum and trimmed mean aggregation rules, respectively. Partial knowledge attacks are considered. The numbers are testing error rates.

	Krum	Trimmed mean	Median
No attack	0.14	0.12	0.13
Krum attack	0.70	0.15	0.18
Trimmed mean attack	0.14	0.25	0.20

Comparing with Back-gradient Optimization based Attack

- Single worker. In this scenario, the attacker distributes the poisoned data on a single compromised worker device.
- Uniform distribution. In this scenario, the attacker distributes the poisoned data to the compromised worker devices uniformly at random.
- BGA has limited success at attacking Byzantine-robust aggregation rules, while our attacks can substantially increase the testing error rates
- if the federated learning uses the mean aggregation rule BGA is still successful
- when applying our attacks for trimmed mean to attack the mean aggregation rule, we can increase the testing error rates substantially more

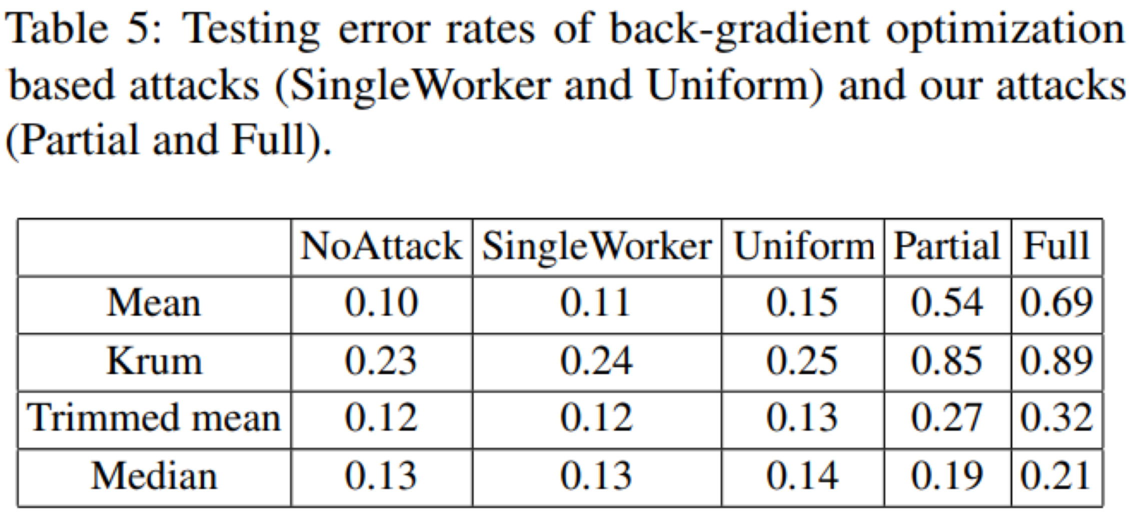
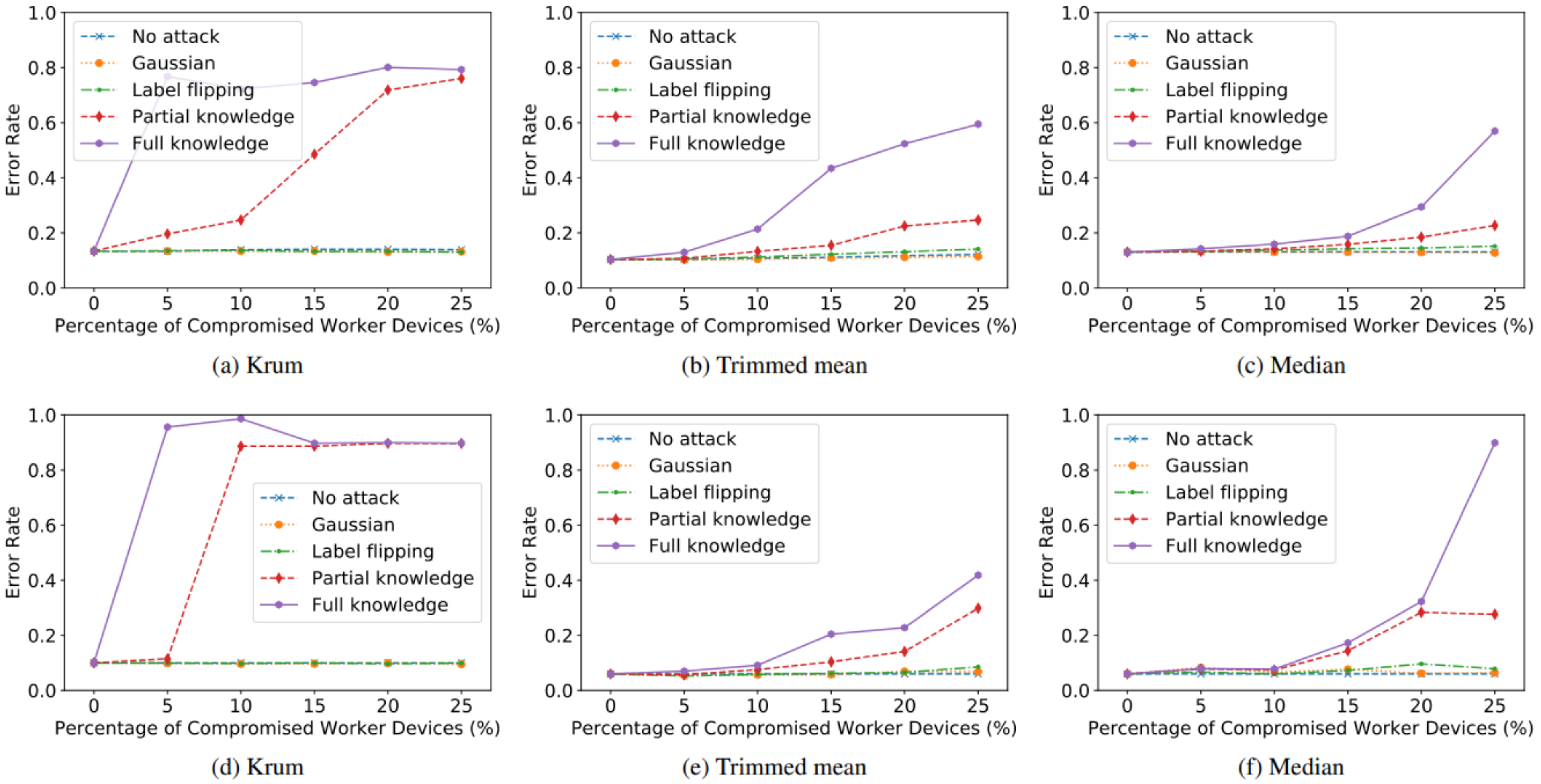


Table 5: Testing error rates of back-gradient optimization based attacks (SingleWorker and Uniform) and our attacks (Partial and Full).

	NoAttack	SingleWorker	Uniform	Partial	Full
Mean	0.10	0.11	0.15	0.54	0.69
Krum	0.23	0.24	0.25	0.85	0.89
Trimmed mean	0.12	0.12	0.13	0.27	0.32
Median	0.13	0.13	0.14	0.19	0.21

Effects of # compromised devices



(a) Krum

(b) Trimmed mean

(c) Median

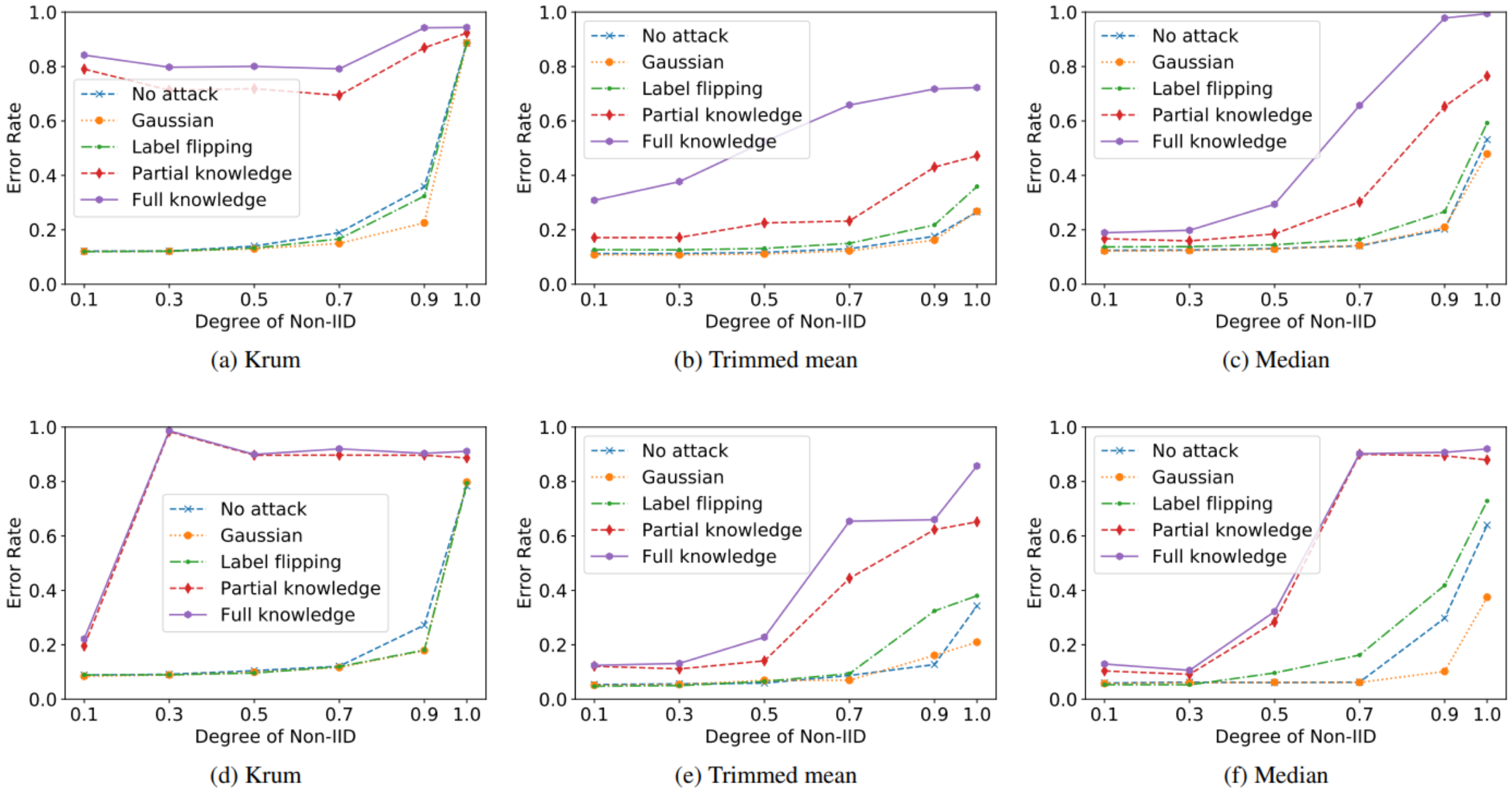
(d) Krum

(e) Trimmed mean

(f) Median

Figure 2: Testing error rates for different attacks as we have more compromised worker devices on MNIST. (a)-(c): LR classifier and (d)-(f): DNN classifier.

Effects of sample correlation (non-IID)



(a) Krum

(b) Trimmed mean

(c) Median

(d) Krum

(e) Trimmed mean

(f) Median

Figure 3: Testing error rates for different attacks as we increase the degree of non-IID on MNIST. (a)-(c): LR classifier and (d)-(f): DNN classifier.

- Error rates of all attacks including no attacks increase as we increase the degree of non-IID, except that the error rates of our attacks to Krum fluctuate as the degree of non-IID increases
- the local training datasets on different worker devices are more non-IID, the local models are more diverse, leaving more room for attacks

RONI and TRIM

- RONI and TRIM are defense algorithm agist data poisoning attack
- RONI: remove samples with large negative impact on **error rate**
- TRIM: remove samples with large negative impact on **loss value**

Adaptive defender:

Generalize RONI and TRIM to reject “abnormal weight” by having the central server holding a validation set to discard abnormal weight.

Error Rate (ERR) and Loss Function(LFR) based Rejection

Cost:

Longer training time to adapt new samples

Longer aggregation time.

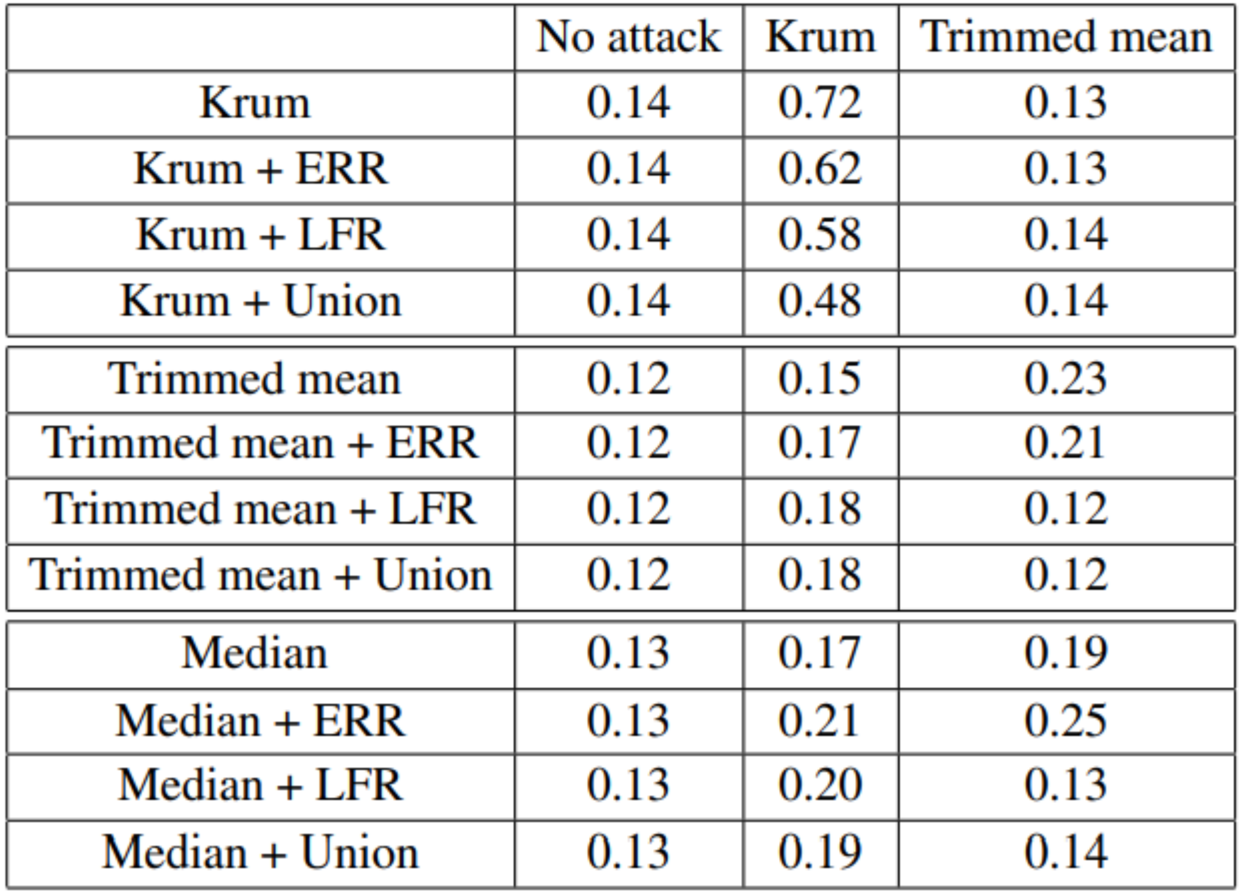
Benefit:

Reduces error rate in all cases

Some even fully suppress the effects from adversarial weights

Defense performance

- On MNIST dataset
- Krum is still vulnerable if the attack rule is known
- LFR and Union are generally more effective than ERR defense



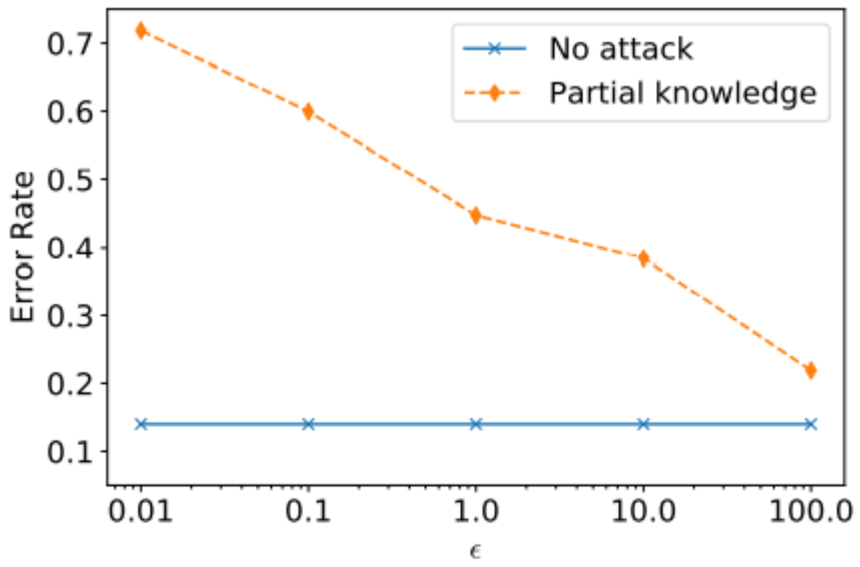
	No attack	Krum	Trimmed mean
Krum	0.14	0.72	0.13
Krum + ERR	0.14	0.62	0.13
Krum + LFR	0.14	0.58	0.14
Krum + Union	0.14	0.48	0.14
Trimmed mean	0.12	0.15	0.23
Trimmed mean + ERR	0.12	0.17	0.21
Trimmed mean + LFR	0.12	0.18	0.12
Trimmed mean + Union	0.12	0.18	0.12
Median	0.13	0.17	0.19
Median + ERR	0.13	0.21	0.25
Median + LFR	0.13	0.20	0.13
Median + Union	0.13	0.19	0.14

Strength:

- The first systematic study on local model poisoning attacks to Byzantine-robust federated learning.
- The proposed local model poisoning attacks is more powerful than any other attacking methods to federated learning.
- The impact of different parameters on the attacking performance is systematically analyzed.

Limitation:

- We don’t know if the attack will remain successful on models with more parameters such as ResNets…
- Attack on Krum has similar flavor to the attack against recommender systems. The Distances between adversarial models may be closed than between benign models.



(b)