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]*IBM Research*

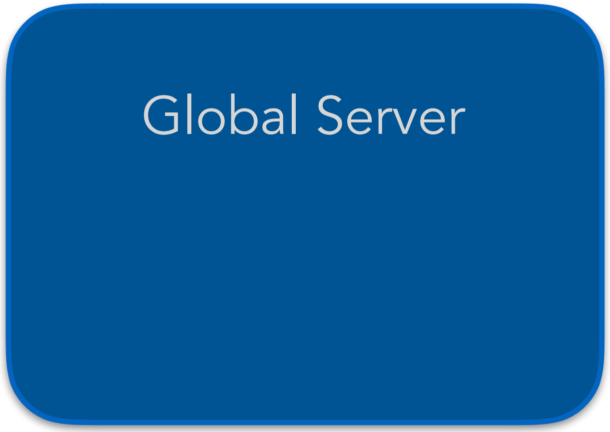
ICML 2019

Federated learning (with a malicious agent)

Federated learning (with a malicious agent)

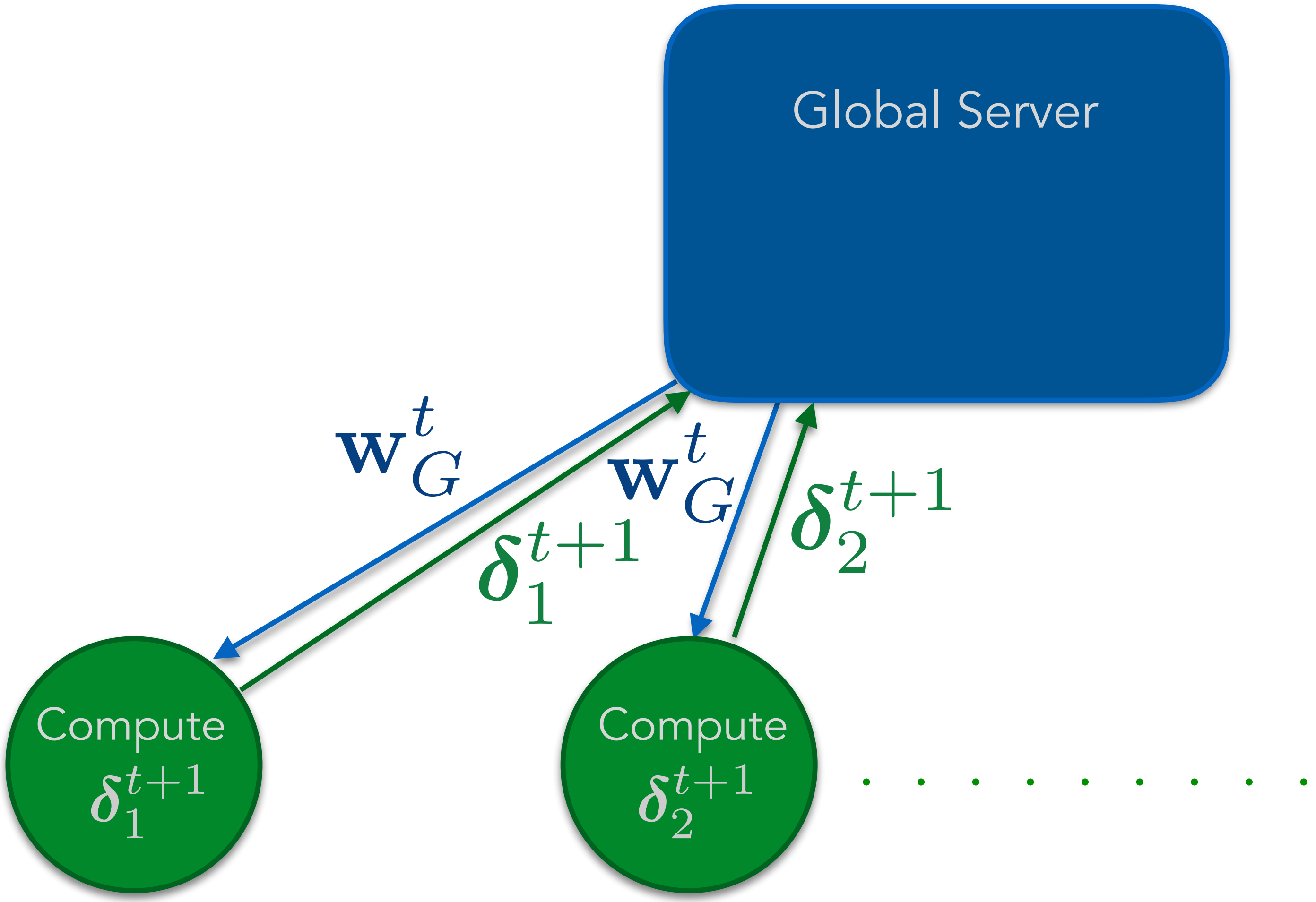
McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS 2017

Federated learning (with a malicious agent)



McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS 2017

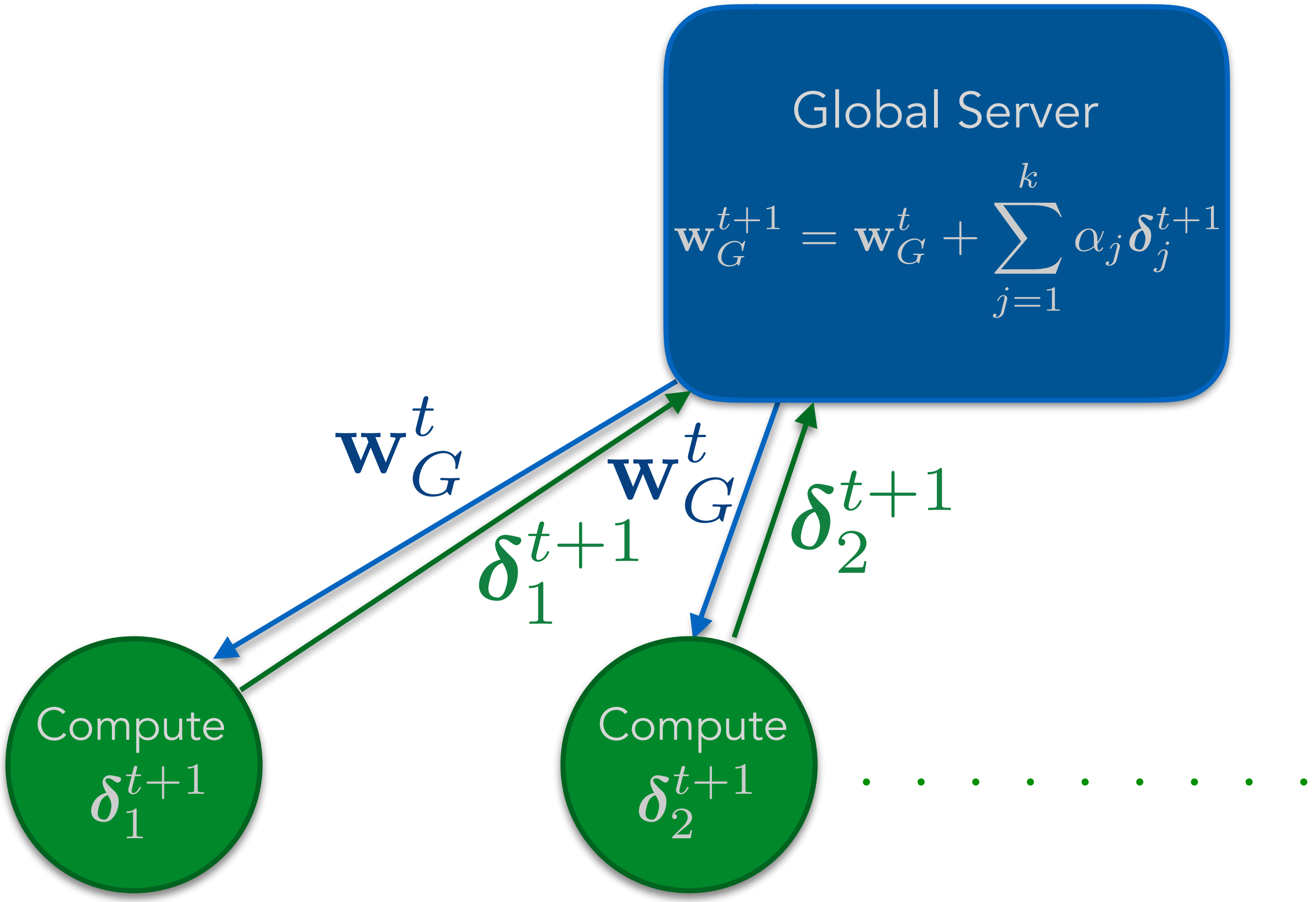
Federated learning (with a malicious agent)



$$\forall j \neq m,\ \boldsymbol{\delta}_j^{t+1} = \operatorname*{argmin}_{\boldsymbol{\delta}} L_{\text{train}}\left(\{\mathbf{x}_j^i, y_j^i\}_{i=1}^{n_j}; \mathbf{w}_G^t + \boldsymbol{\delta}\right)$$

McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS 2017

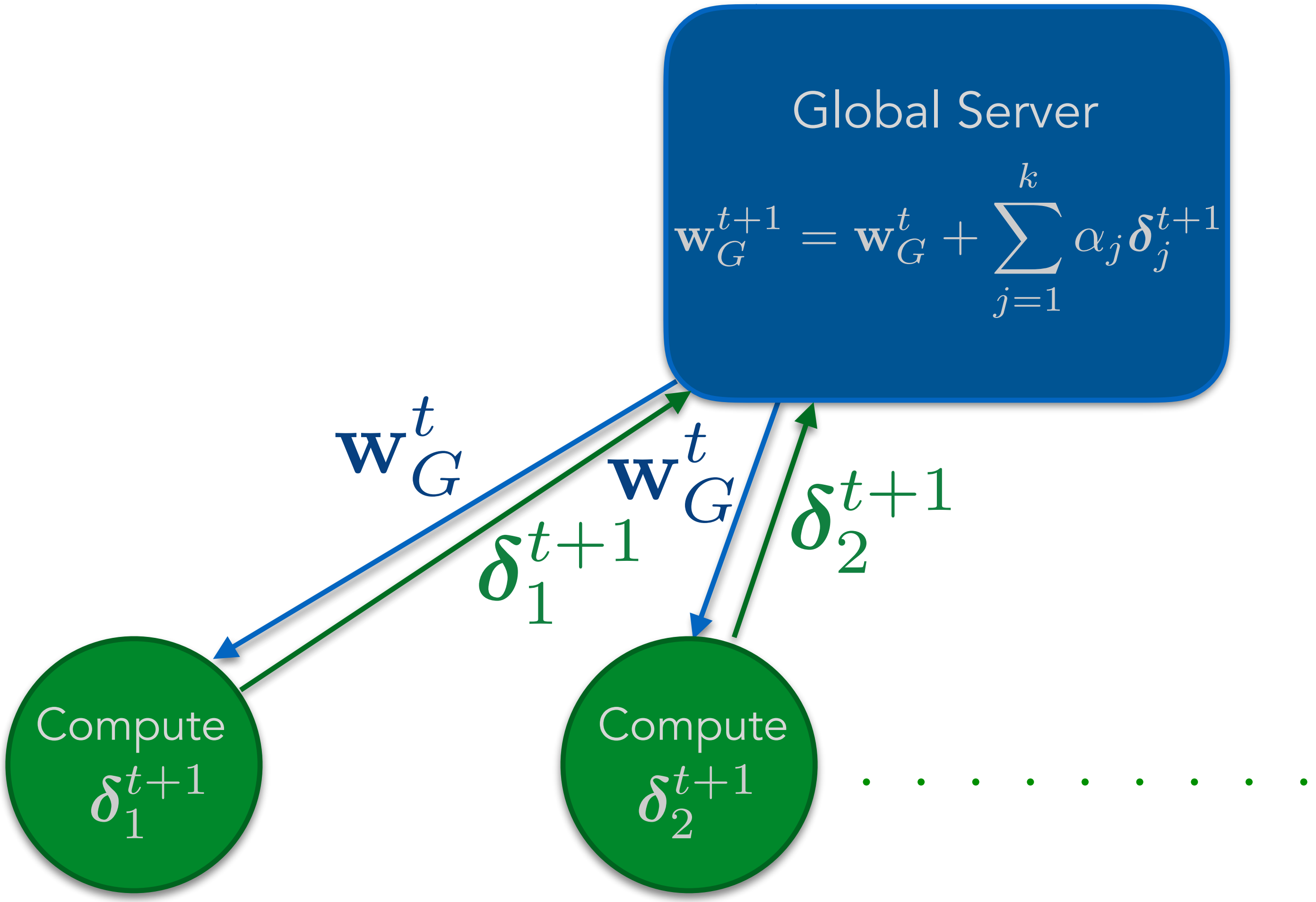
Federated learning (with a malicious agent)



$$\forall j \neq m,\ \boldsymbol{\delta}_j^{t+1} = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, L_{\text{train}}\left(\{\mathbf{x}_j^i, y_j^i\}_{i=1}^{n_j}; \mathbf{w}_G^t + \boldsymbol{\delta}\right)$$

McMahan et al., *Communication-Efficient Learning of Deep Networks from Decentralized Data*, AISTATS 2017

Federated learning (with a malicious agent)



$$\forall j \neq m,\ \boldsymbol{\delta}_j^{t+1} = \operatorname*{argmin}_{\boldsymbol{\delta}} L_{\text{train}} \left(\{\mathbf{x}_j^i, y_j^i\}_{i=1}^{n_j}; \mathbf{w}_G^t + \boldsymbol{\delta} \right)$$

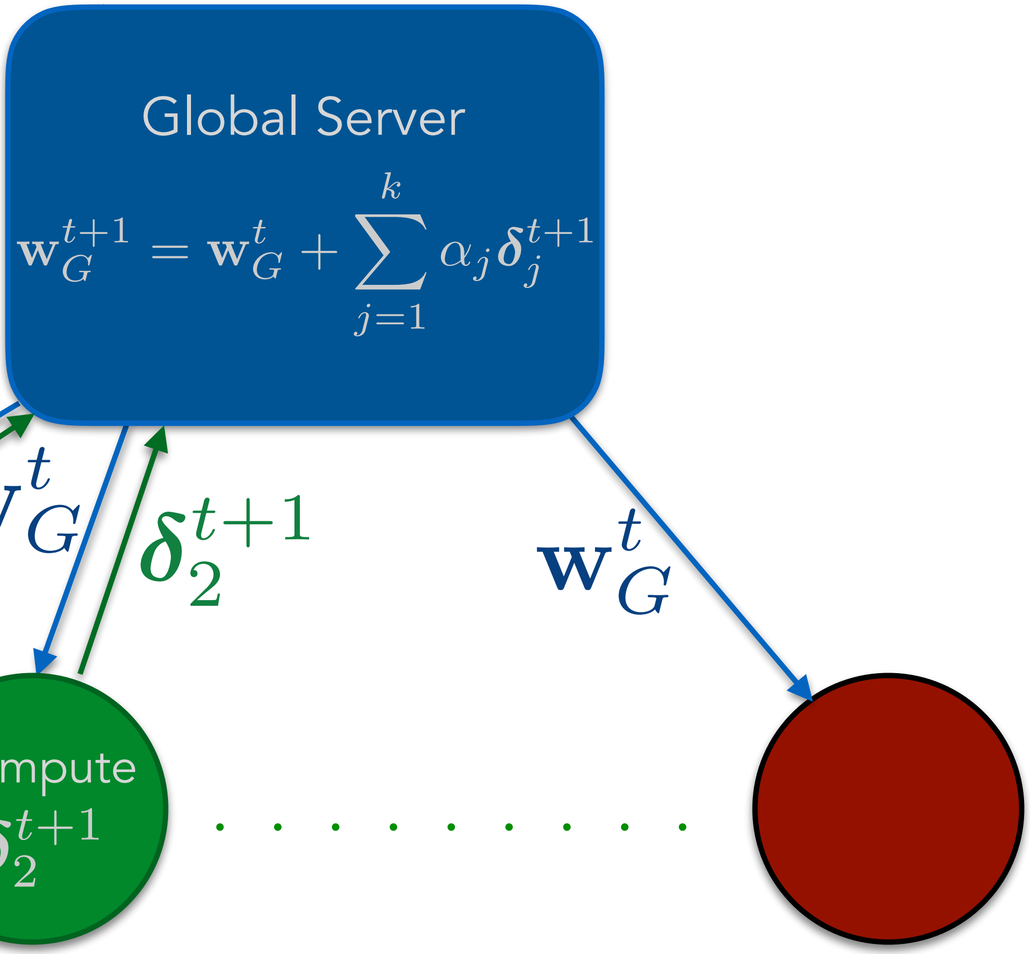
Federated learning (with a malicious agent)

Threat model

- Single malicious agent

Information available:

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



$$\forall j \neq m,\ \boldsymbol{\delta}_j^{t+1} = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, L_{\text{train}} \left(\{\mathbf{x}_j^i, y_j^i\}_{i=1}^{n_j}; \mathbf{w}_G^t + \boldsymbol{\delta} \right)$$

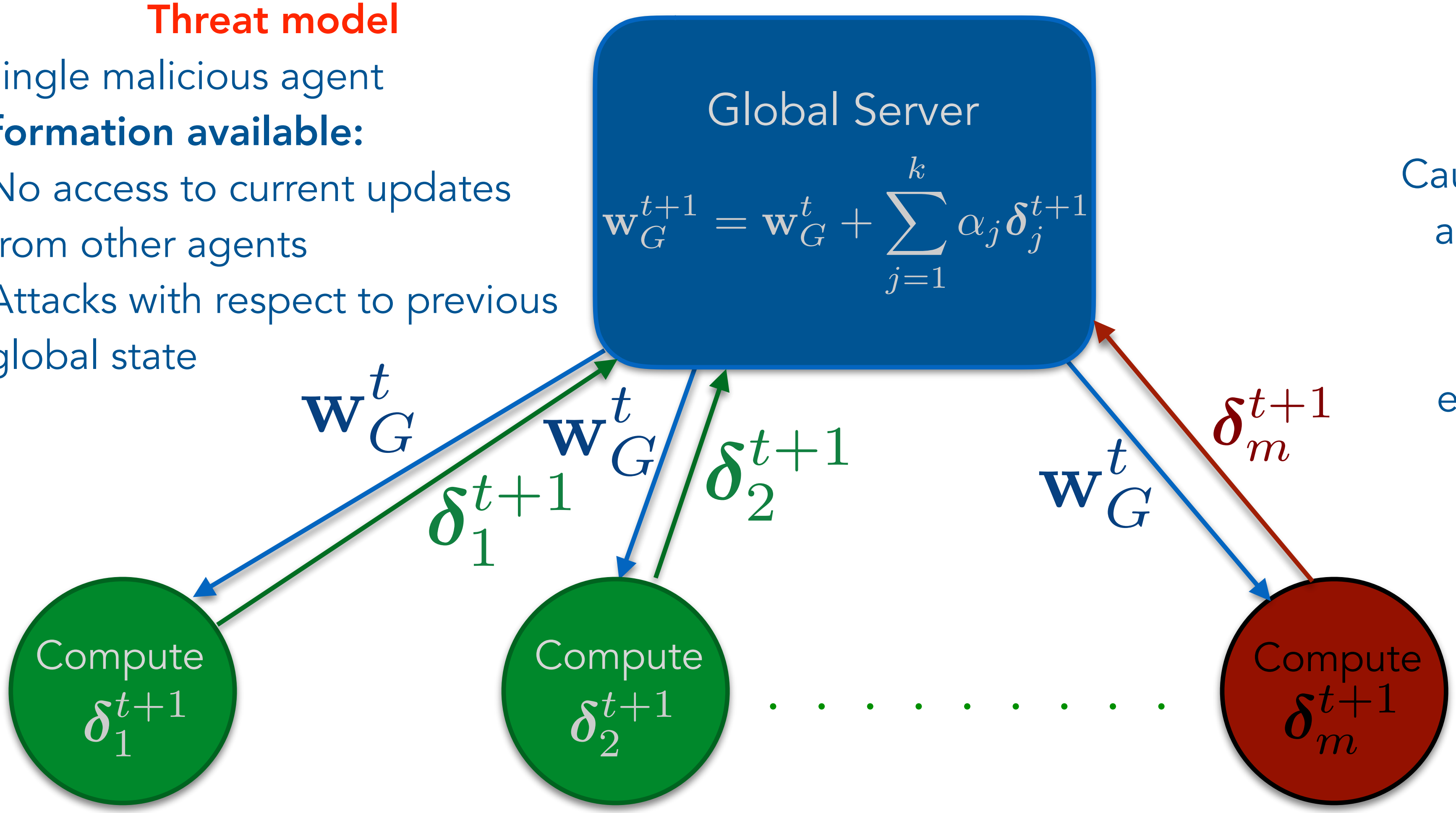
Federated learning (with a malicious agent)

Threat model

- Single malicious agent

Information available:

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



Aim

Cause targeted misclassification of an auxiliary set of examples for the global model

and

ensure global model has good performance

$\forall j \neq m,\ \boldsymbol{\delta}_j^{t+1} = \underset{\boldsymbol{\delta}}{\operatorname{argmin}}\, L_{\text{train}}\left(\{\mathbf{x}_j^i, y_j^i\}_{i=1}^{n_j}; \mathbf{w}_G^t + \boldsymbol{\delta}\right)$

$\boldsymbol{\delta}_m^{t+1} = \mathcal{A}\left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n_m}, \{\mathbf{x}^l, T^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G^t + \boldsymbol{\delta}\right)$

Targeted Model Poisoning

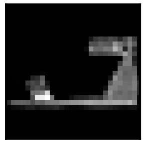
Targeted Model Poisoning

Strategy	**Malicious agent's update computation**
Boosting malicious update, no local training	$\boldsymbol{\delta}_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}_m^l, T_m^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ $\boldsymbol{\delta}_{\text{mal}} \rightarrow \beta \boldsymbol{\delta}_{\text{mal}}$

Targeted Model Poisoning

Strategy	Malicious agent's update computation
Boosting malicious update, no local training	$\boldsymbol{\delta}_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}_m^l, T_m^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ $\boldsymbol{\delta}_{\text{mal}} \to \beta \boldsymbol{\delta}_{\text{mal}}$

Evaluation setup

- Fashion MNIST data [2]
- CNN achieving 91.5% accuracy on test data
- Total of **10 agents**, all called every time step
- Training is stopped when global model achieves above 91% validation accuracy
- **Adversarial objective**: Classify  ('sandal', class 5) as a 'sneaker', class 7

Targeted Model Poisoning

Strategy	Malicious agent's update computation
Boosting malicious update, no local training	$\boldsymbol{\delta}_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}_m^l, T_m^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ $\boldsymbol{\delta}_{\text{mal}} \rightarrow \beta \boldsymbol{\delta}_{\text{mal}}$

- Adam for 5 epochs
- Boosting by 10

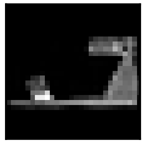
Evaluation setup

✦ Fashion MNIST data [2]

✦ CNN achieving 91.5% accuracy on test data

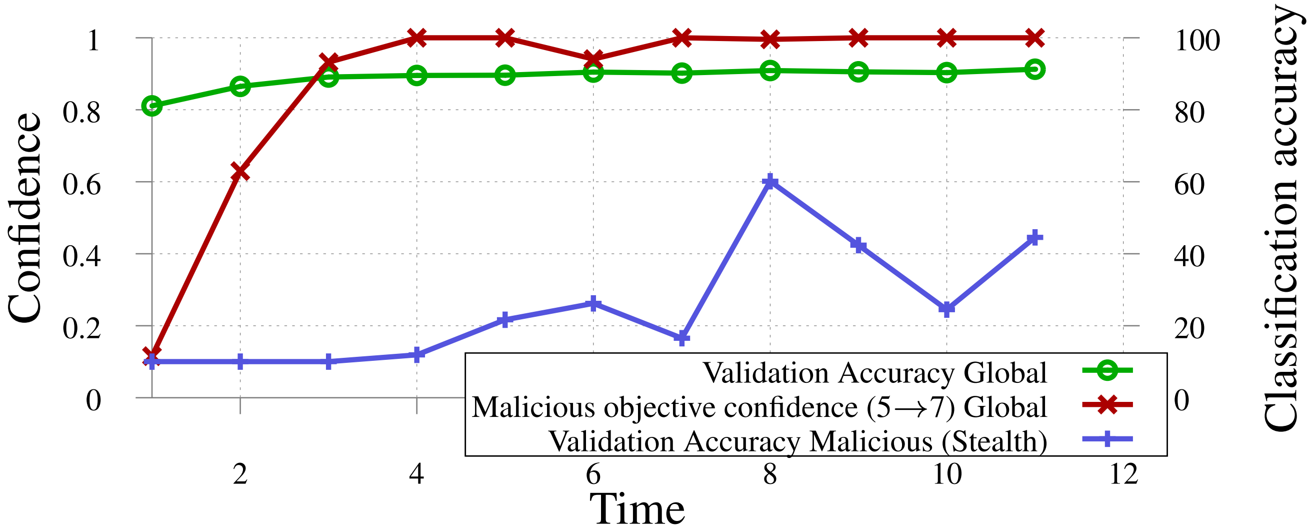
✦ Total of **10 agents**, all called every time step

✦ Training is stopped when global model achieves above 91% validation accuracy

✦ **Adversarial objective**: Classify  ('sandal', class 5) as a 'sneaker', class 7

Targeted Model Poisoning: *Results*

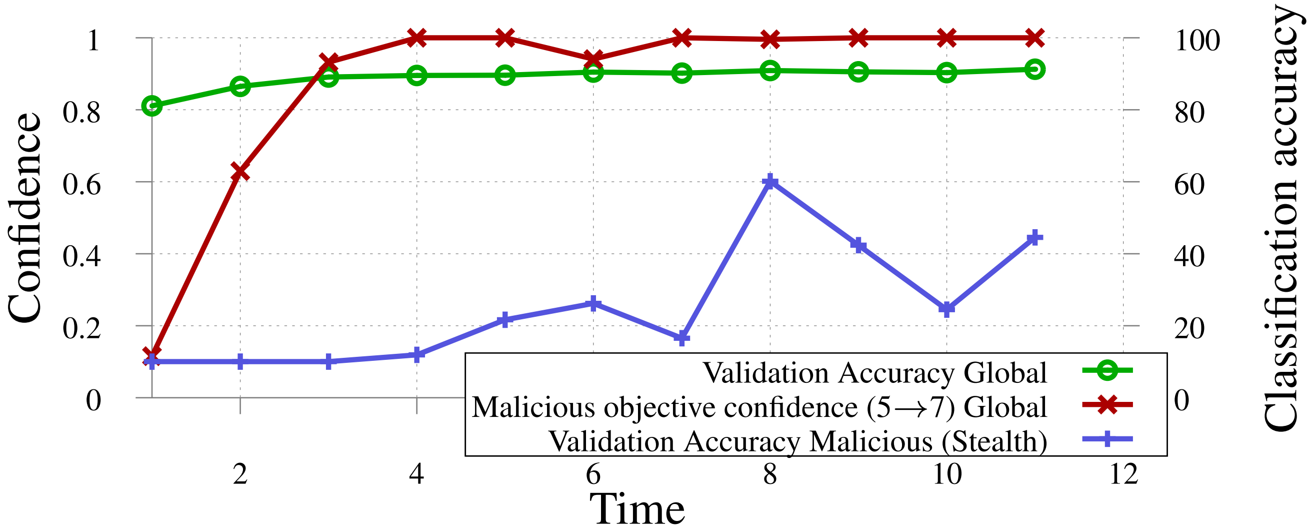
Targeted Model Poisoning: *Results*

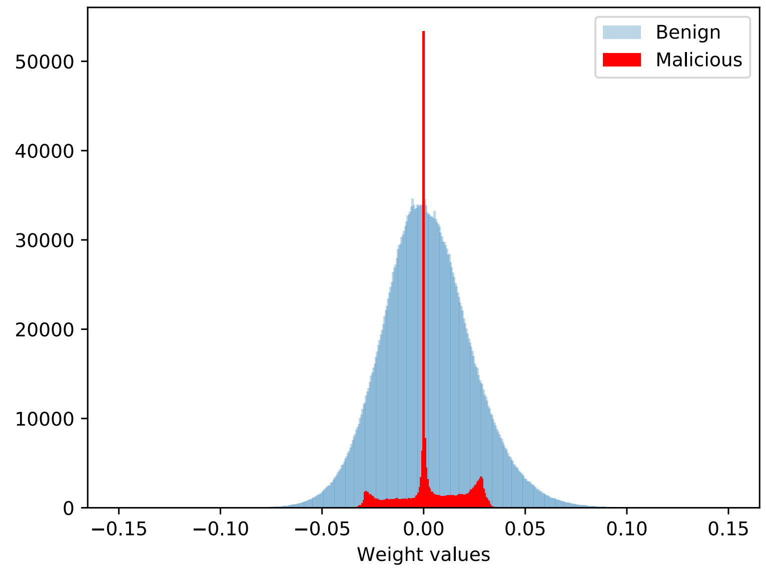


Takeaways

1. Targeted backdoor inserted with high confidence
2. Accuracy on validation data does not suffer for global model
3. Malicious model has low validation accuracy

Targeted Model Poisoning: *Results*





Takeaways

1. Targeted backdoor inserted with high confidence
2. Accuracy on validation data does not suffer for global model
3. Malicious model has low validation accuracy

Takeaways

1. Weight update distributions for benign and malicious agents are very different
2. Malicious update could be 'hidden' inside benign one

Targeted Model Poisoning: Alternating Minimization attack

Targeted Model Poisoning: Alternating Minimization attack

Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	

Targeted Model Poisoning: Alternating Minimization attack

Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	$\boldsymbol{\delta}'_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}_m^l, T_m^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ Malicious Objective

Targeted Model Poisoning: Alternating Minimization attack

Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	$\boldsymbol{\delta}'_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}^l_m, T^l_m\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ Malicious Objective $\boldsymbol{\delta}'_{\text{mal}} \rightarrow \beta \boldsymbol{\delta}'_{\text{mal}}$

Targeted Model Poisoning: Alternating Minimization attack

Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	$\boldsymbol{\delta}'_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}^l_m, T^l_m\}^{n_{\text{mal}}}_{l=1}; \mathbf{w}_G + \boldsymbol{\delta})$ Malicious Objective $\boldsymbol{\delta}'_{\text{mal}} \rightarrow \beta\boldsymbol{\delta}'_{\text{mal}}$ $\boldsymbol{\delta}''_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}\left(\{\mathbf{x}^i_m, y^i_m\}^n_{i=1}; \mathbf{w}_G + \beta\boldsymbol{\delta}'_{\text{mal}} + \boldsymbol{\delta}\right) + \rho\|\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{cons}}\|^2_2$ Benign Objective Distance Constraint

Targeted Model Poisoning: Alternating Minimization attack

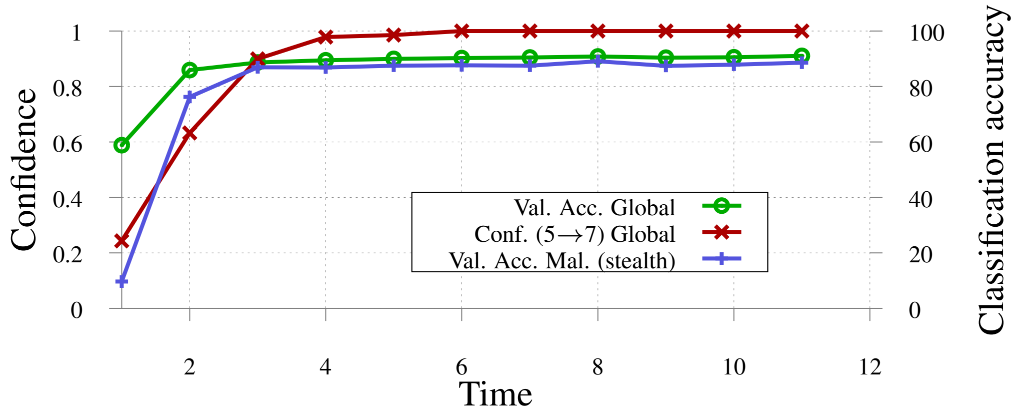
Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	*Repeat:* $\boldsymbol{\delta}'_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}_m^l, T_m^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ Malicious Objective $\boldsymbol{\delta}'_{\text{mal}} \to \beta \boldsymbol{\delta}'_{\text{mal}}$ $\boldsymbol{\delta}''_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}\left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n}; \mathbf{w}_G + \beta \boldsymbol{\delta}'_{\text{mal}} + \boldsymbol{\delta}\right) + \rho \lVert \boldsymbol{\delta} - \boldsymbol{\delta}_{\text{cons}} \rVert_2^2$ Benign Objective Distance Constraint

For every step w.r.t. to the malicious loss, take 10 steps for the benign loss

Targeted Model Poisoning: Alternating Minimization attack

Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	$\delta'_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \text{Cross-entropy}(\{\mathbf{x}_m^l, T_m^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta})$ Malicious Objective *Repeat:* $\boldsymbol{\delta}'_{\text{mal}} \rightarrow \beta\boldsymbol{\delta}'_{\text{mal}}$ $\boldsymbol{\delta}''_{\text{mal}} = \text{argmin}_{\boldsymbol{\delta}} \underbrace{\text{Cross-entropy}\left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n}; \mathbf{w}_G + \beta\boldsymbol{\delta}'_{\text{mal}} + \boldsymbol{\delta}\right)}_{\text{Benign Objective}} + \underbrace{\rho\|\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{cons}}\|_2^2}_{\text{Distance Constraint}}$

For every step w.r.t. to the malicious loss, take 10 steps for the benign loss



Takeaway

Malicious objective is met while maintaining high validation accuracy for malicious model

Targeted Model Poisoning: Alternating Minimization attack

Strategy	Malicious agent's update computation
Alternating minimization of benign and malicious objectives, with distance constraints	$\delta'_{\text{mal}} = \text{argmin}_{\delta}\text{Cross-entropy}(\{\mathbf{x}^l_m, T^l_m\}^{n_{\text{mal}}}_{l=1}; \mathbf{w}_G + \delta)$ Malicious Objective

Repeat:

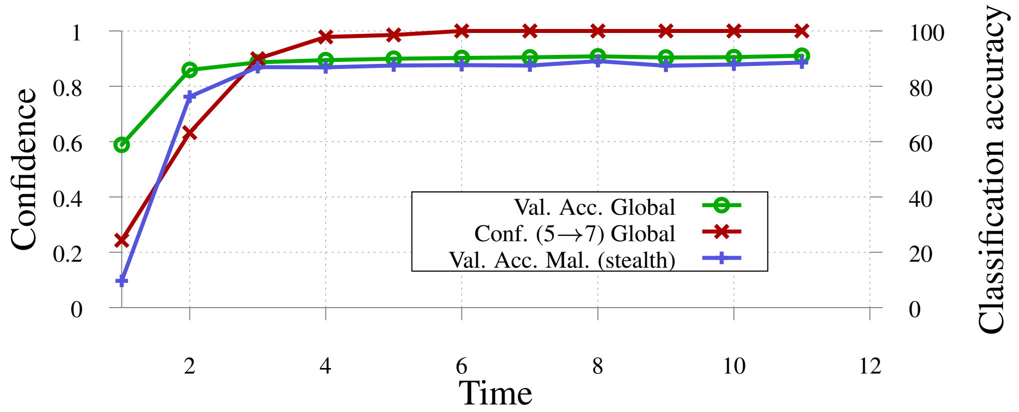
For every step w.r.t. to the malicious loss, take 10 steps for the benign loss

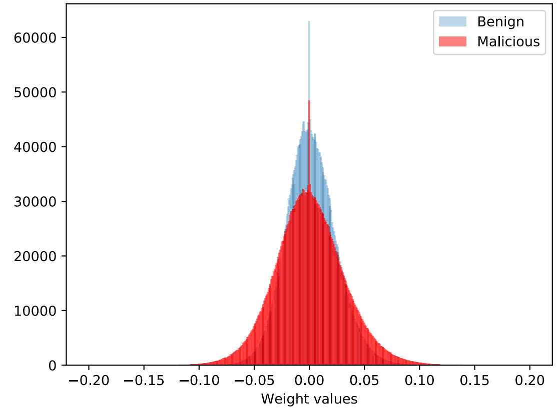
$$\delta'_{\text{mal}} \rightarrow \beta\delta'_{\text{mal}}$$

$$\delta''_{\text{mal}} = \text{argmin}_{\delta}\ \text{Cross-entropy}\left(\{\mathbf{x}^i_m, y^i_m\}^n_{i=1}; \mathbf{w}_G + \beta\delta'_{\text{mal}} + \delta\right) + \rho\|\delta - \delta_{\text{cons}}\|^2_2$$

Benign Objective

Distance Constraint





Takeaway

Malicious objective is met while maintaining high validation accuracy for malicious model

Takeaway

Shape and range match closely due to distance constraint

In summary…

More details and results in our poster (#144 tonight in the Pacific Ballroom)

- Quantitative weight update statistics-based stealth results
- Attacks on Byzantine-resilient aggregation mechanisms
- Connections between model poisoning and interpretability

In summary…

- Federated learning is vulnerable to model poisoning attacks

More details and results in our poster (#144 tonight in the Pacific Ballroom)

- Quantitative weight update statistics-based stealth results
- Attacks on Byzantine-resilient aggregation mechanisms
- Connections between model poisoning and interpretability

In summary…

- Federated learning is vulnerable to model poisoning attacks
- Detection strategies make attacks more challenging, but can be overcome by white-box attackers

More details and results in our poster (#144 tonight in the Pacific Ballroom)

- Quantitative weight update statistics-based stealth results
- Attacks on Byzantine-resilient aggregation mechanisms
- Connections between model poisoning and interpretability

In summary…

- Federated learning is vulnerable to model poisoning attacks
- Detection strategies make attacks more challenging, but can be overcome by white-box attackers
- **Open research question**: Can we develop distributed learning algorithms robust to model poisoning attacks?

More details and results in our poster (#144 tonight in the Pacific Ballroom)

- Quantitative weight update statistics-based stealth results
- Attacks on Byzantine-resilient aggregation mechanisms
- Connections between model poisoning and interpretability

Collaborators

PRINCETON UNIVERSITY

IBM Research

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

Thank you for listening!

Backup slides

Adversarial challenges

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

   Approach: Generate malicious update with respect to $\mathbf{w}_G^t$, i.e. assume $\mathbf{w}_G^t \approx \mathbf{w}_G^{t+1}$

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

   Approach: Generate malicious update with respect to $\mathbf{w}_G^t$, i.e. assume $\mathbf{w}_G^t \approx \mathbf{w}_G^{t+1}$

2. **Averaging with other agents**: Updates from other agents could render malicious agent's update ineffective

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

   Approach: Generate malicious update with respect to $\mathbf{w}_G^t$, i.e. assume $\mathbf{w}_G^t \approx \mathbf{w}_G^{t+1}$

2. **Averaging with other agents**: Updates from other agents could render malicious agent's update ineffective

   Approach: Boost malicious update to overcome effect of scaling

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

    Approach: Generate malicious update with respect to $\mathbf{w}_G^t$, i.e. assume $\mathbf{w}_G^t \approx \mathbf{w}_G^{t+1}$

2. **Averaging with other agents**: Updates from other agents could render malicious agent's update ineffective

    Approach: Boost malicious update to overcome effect of scaling

3. **Randomness in choice of agents**: Malicious agent is not chosen in every iteration if large number of agents

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

    Approach: Generate malicious update with respect to $\mathbf{w}_G^t$, i.e. assume $\mathbf{w}_G^t \approx \mathbf{w}_G^{t+1}$

2. **Averaging with other agents**: Updates from other agents could render malicious agent's update ineffective

    Approach: Boost malicious update to overcome effect of scaling

3. **Randomness in choice of agents**: Malicious agent is not chosen in every iteration if large number of agents

4. **Avoid detection**: Server may detect based on effect on accuracy on validation data or weight update statistics

Adversarial challenges

1. **No access to other agents' updates at time *t***: Adversary has no access to current updates from the other agents when attempting model poisoning

   Approach: Generate malicious update with respect to $\mathbf{w}_G^t$, i.e. assume $\mathbf{w}_G^t \approx \mathbf{w}_G^{t+1}$

2. **Averaging with other agents**: Updates from other agents could render malicious agent's update ineffective

   Approach: Boost malicious update to overcome effect of scaling

3. **Randomness in choice of agents**: Malicious agent is not chosen in every iteration if large number of agents

4. **Avoid detection**: Server may detect based on effect on accuracy on validation data or weight update statistics

   Approach: Improve on baseline by adding benign training and distance constraints

Stealthy Model Poisoning

Strategy	Malicious agent's update computation
Joint minimization of benign and malicious objectives, with distance constraints	$\boldsymbol{\delta}_{\text{mal}} = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \underbrace{L\left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n_m}; \mathbf{w}_G + \boldsymbol{\delta}\right)}_{\text{Benign Objective}} + \underbrace{\beta L\left(\{\mathbf{x}^l, T^l\}_{l=1}^{n_{\text{mal}}}; \mathbf{w}_G + \boldsymbol{\delta}\right)}_{\text{Malicious Objective}} + \underbrace{\rho\|\boldsymbol{\delta} - \boldsymbol{\delta}_{\text{cons}}\|_2^2}_{\text{Distance Constraint}}$

Stealthy Model Poisoning

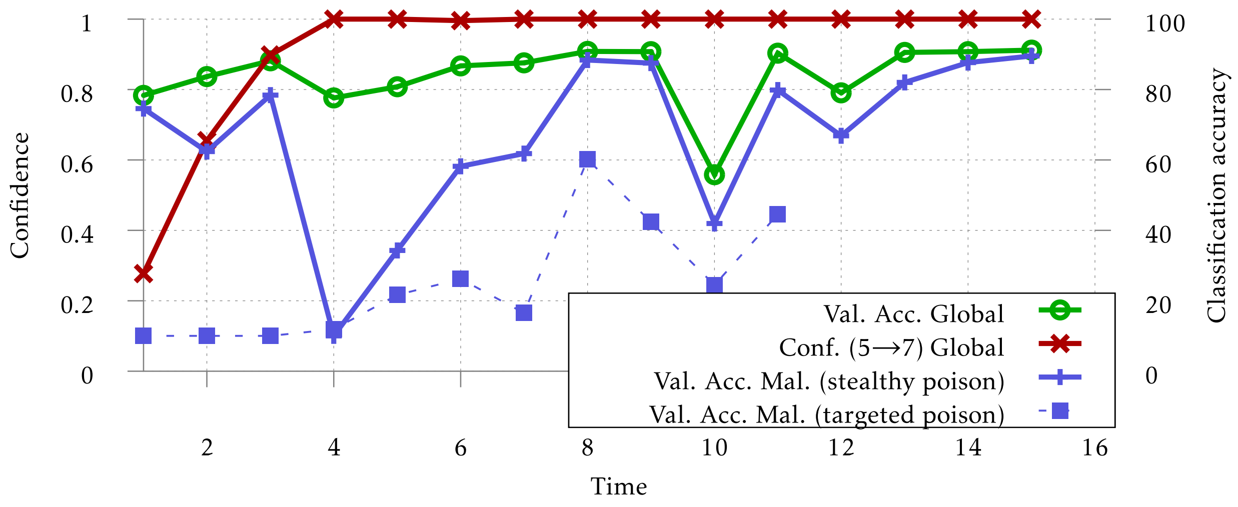
Strategy	Malicious agent's update computation
Joint minimization of benign and malicious objectives, with distance constraints	$\boldsymbol{\delta}_{\mathrm{mal}} = \underset{\boldsymbol{\delta}}{\mathrm{argmin}} \underbrace{L\left(\{\mathbf{x}_m^i, y_m^i\}_{i=1}^{n_m}; \mathbf{w}_G + \boldsymbol{\delta}\right)}_{\text{Benign Objective}} + \underbrace{\beta L\left(\{\mathbf{x}^l, T^l\}_{l=1}^{n_{\mathrm{mal}}}; \mathbf{w}_G + \boldsymbol{\delta}\right)}_{\text{Malicious Objective}} + \underbrace{\rho\|\boldsymbol{\delta} - \boldsymbol{\delta}_{\mathrm{cons}}\|_2^2}_{\text{Distance Constraint}}$

Experiment settings

- Boosting by 10 ($\beta = 10$)
- $\rho = 1e-4$
- Adam for 10 epochs
- Cross-entropy loss
- Constrain w.r.t. previous cumulative update from other agents

Stealthy Model Poisoning: *Results and Weight update*

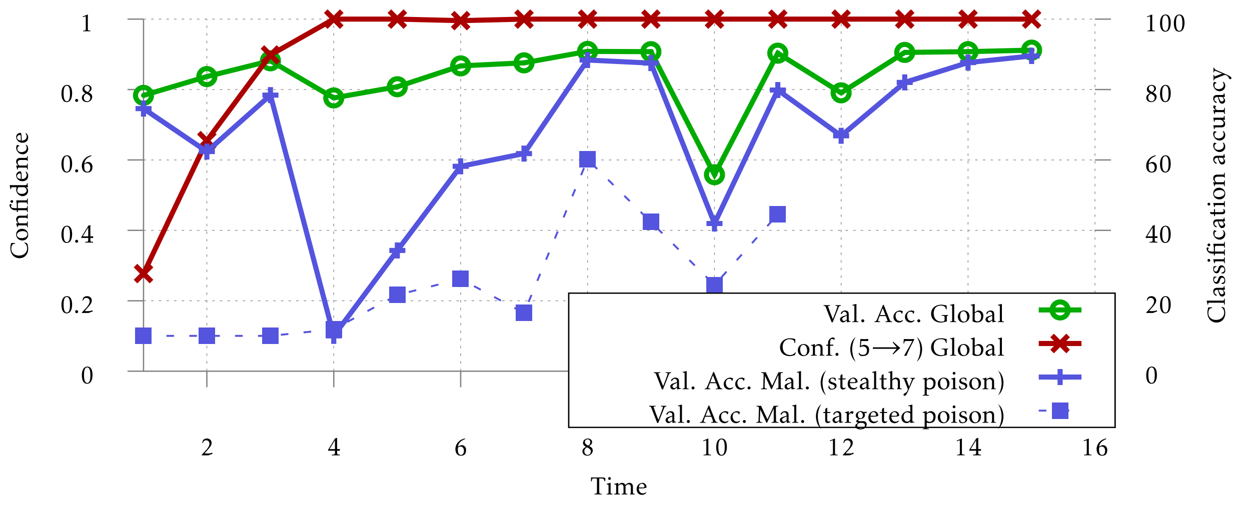
Stealthy Model Poisoning: *Results and Weight update*

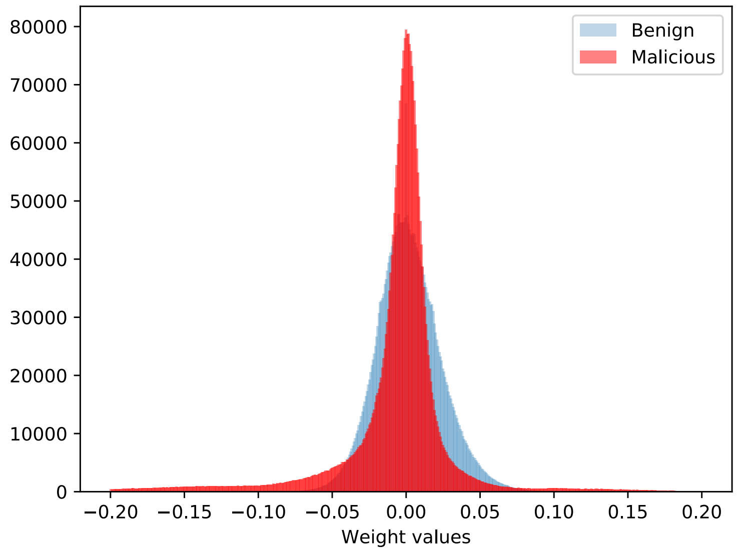


Takeaways

1. Malicious objective is met
2. Improved validation accuracy compared to *Targeted Model Poisoning*

Stealthy Model Poisoning: *Results and Weight update*





Takeaways

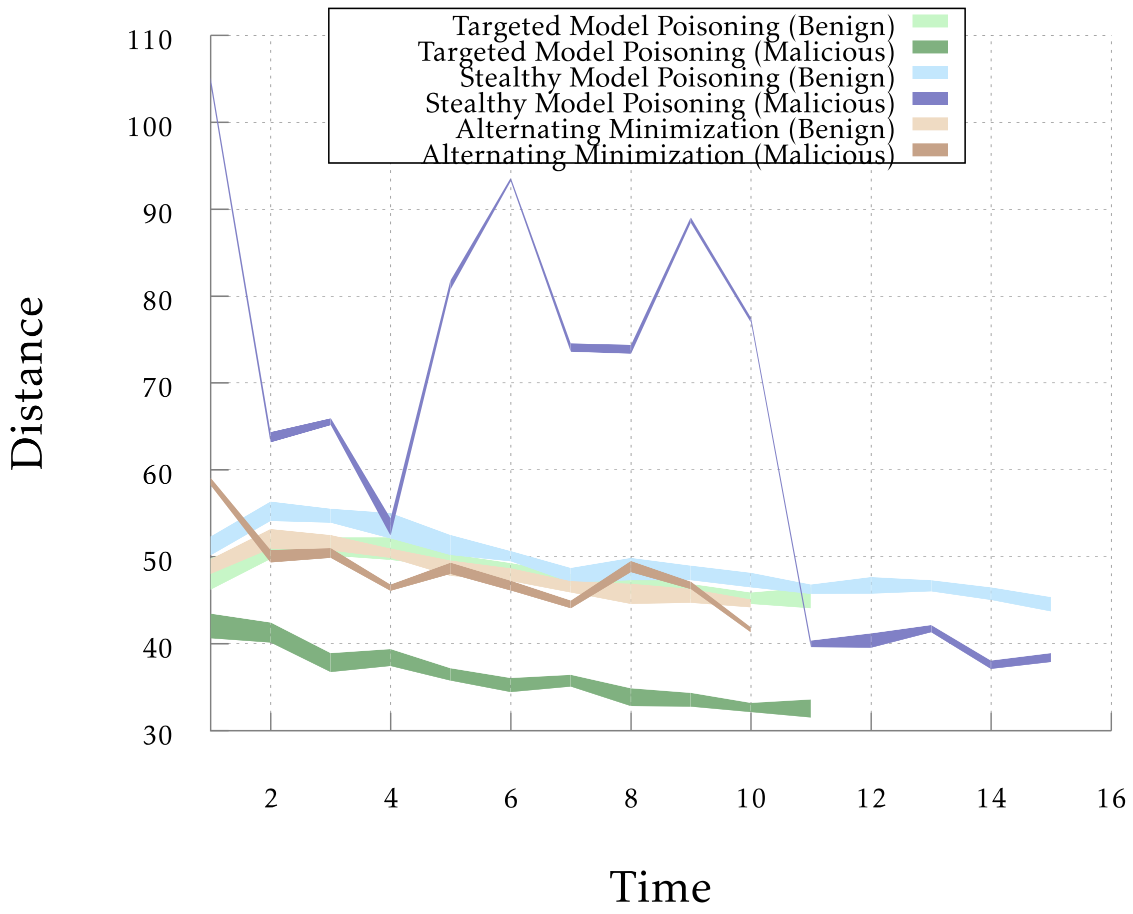
1. Malicious objective is met
2. Improved validation accuracy compared to *Targeted Model Poisoning*

Takeaway

Closer match between weight updates for benign and malicious agents

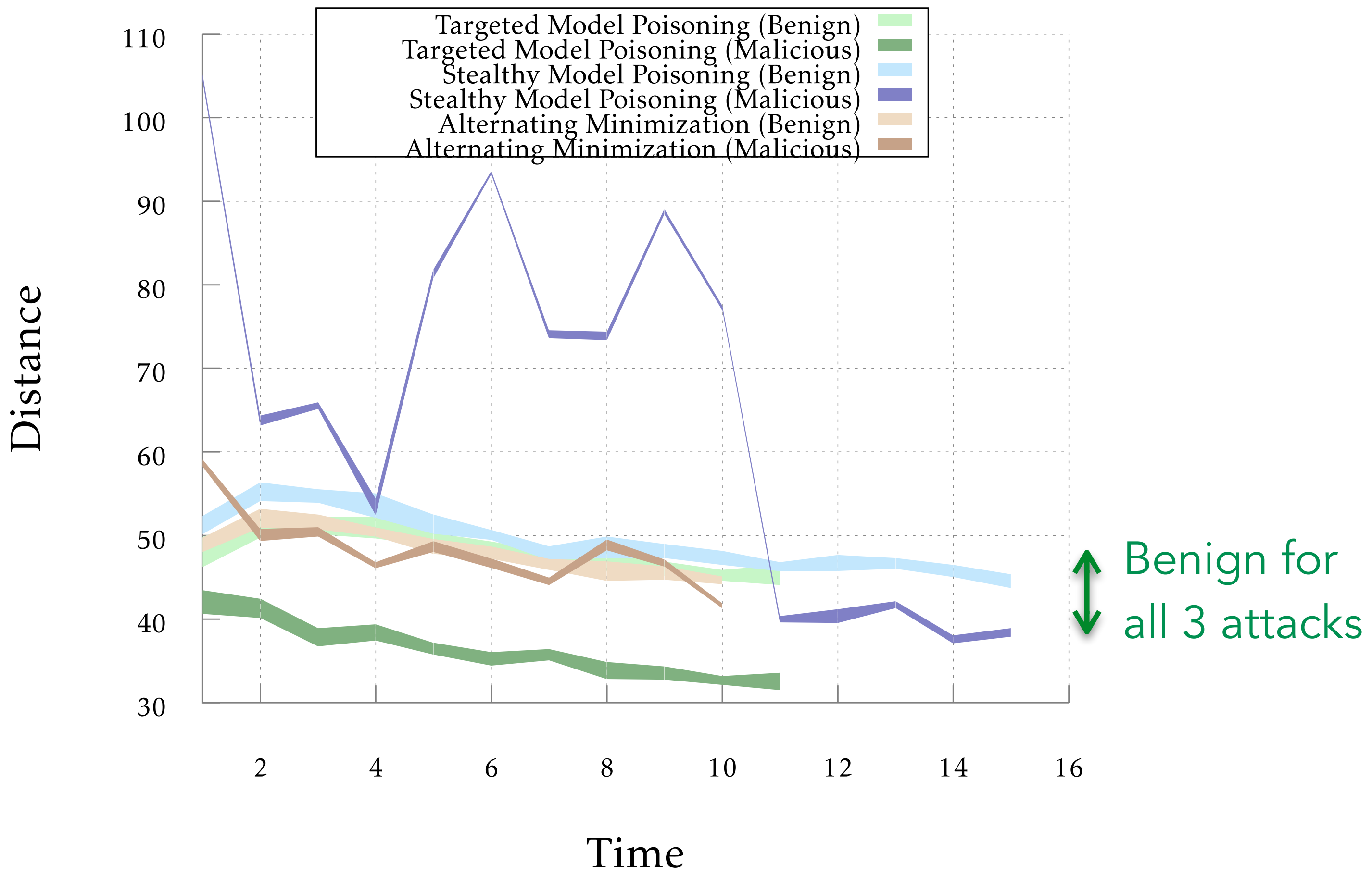
Weight update distance spread (attack stealth measure)

Spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents



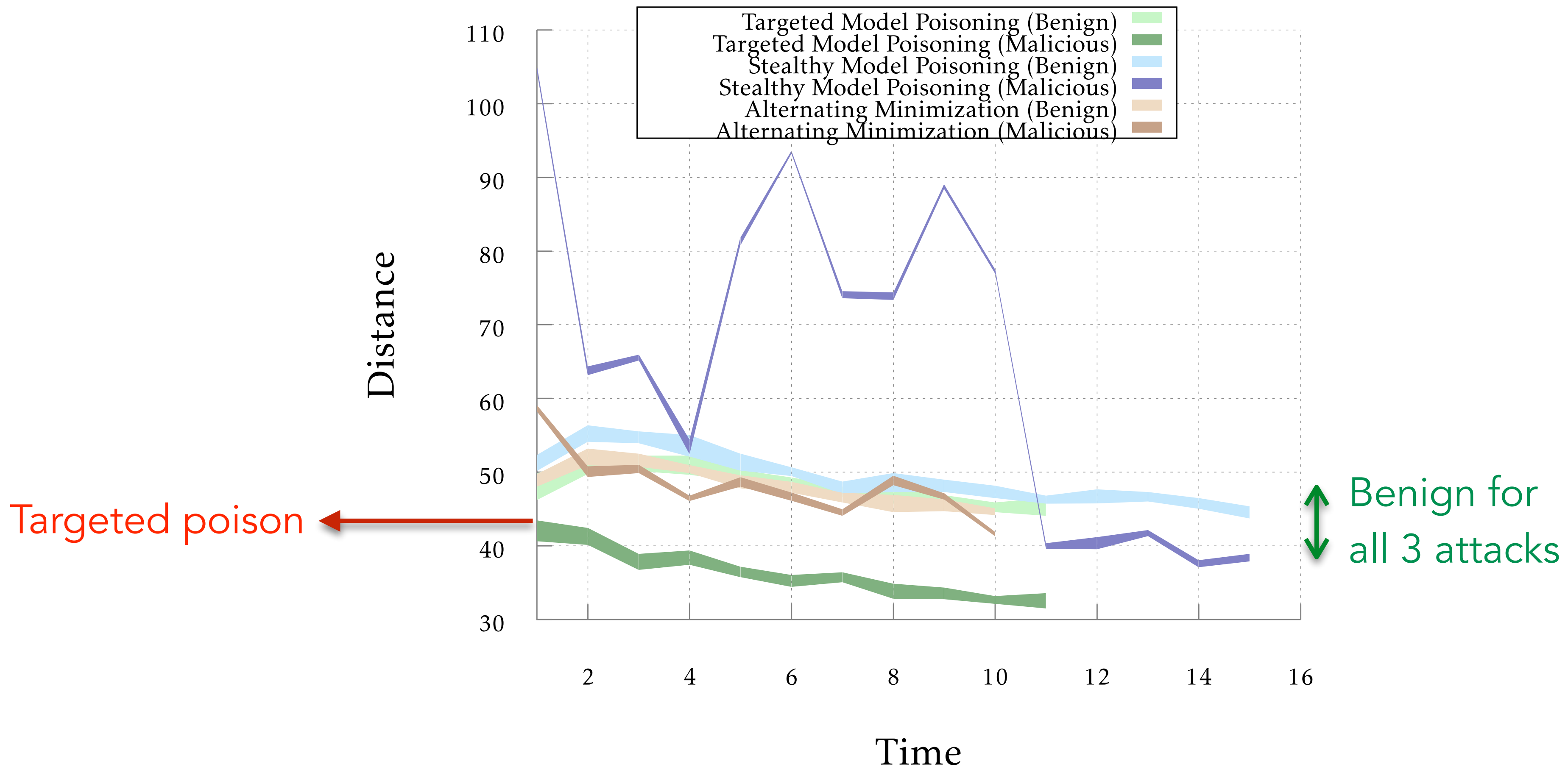
Weight update distance spread (attack stealth measure)

Spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents



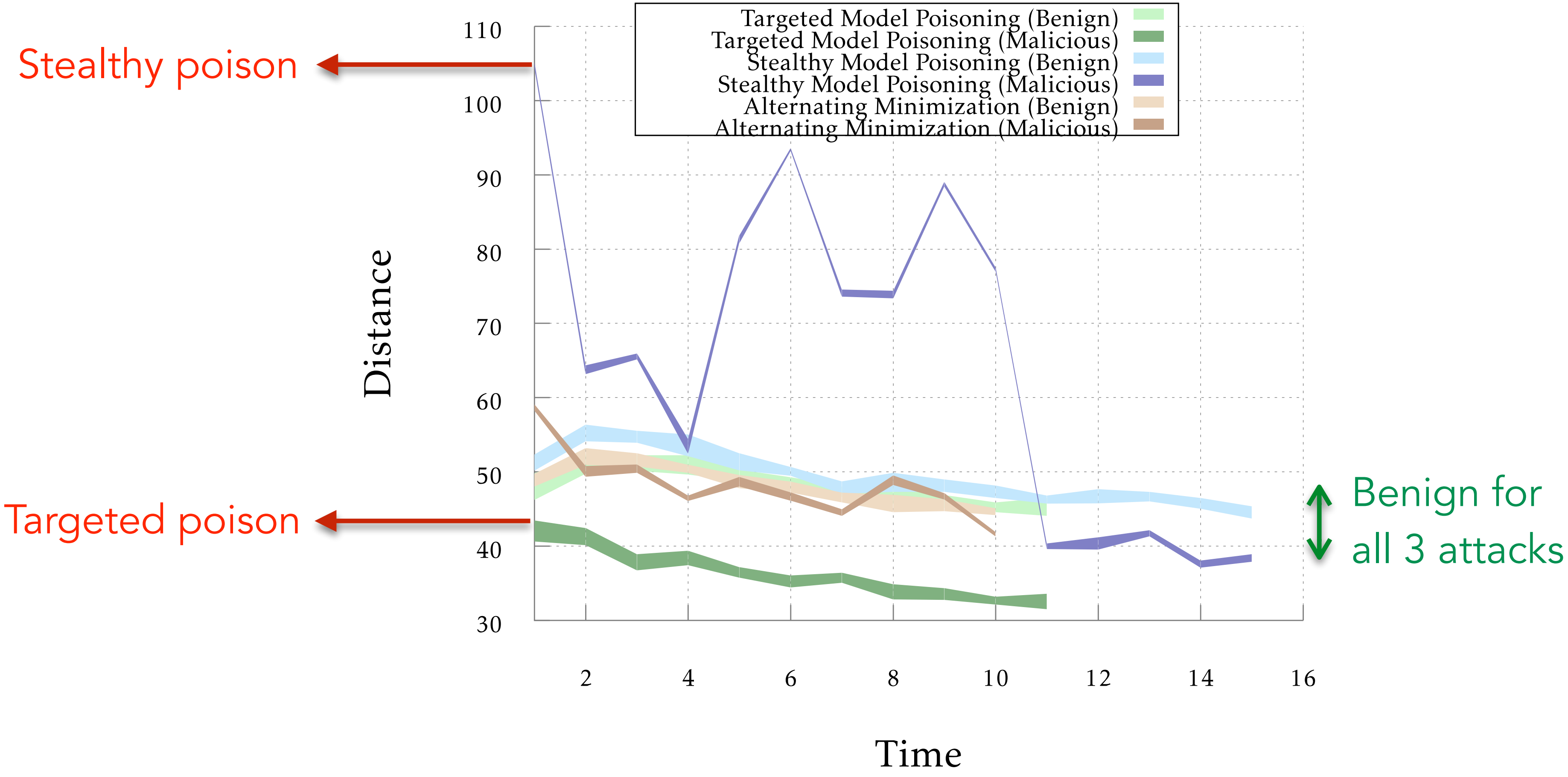
Weight update distance spread (attack stealth measure)

Spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents



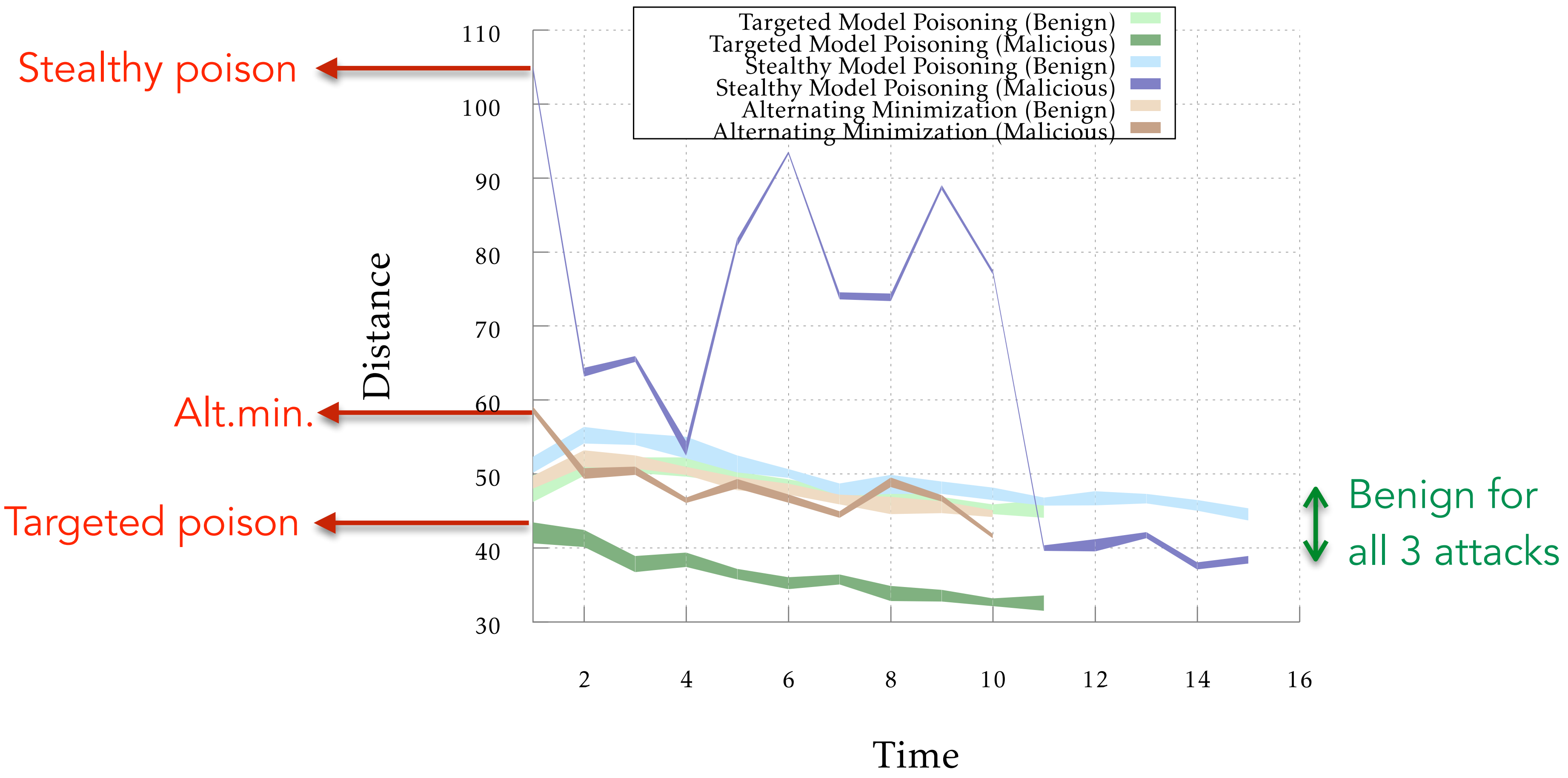
Weight update distance spread (attack stealth measure)

Spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents



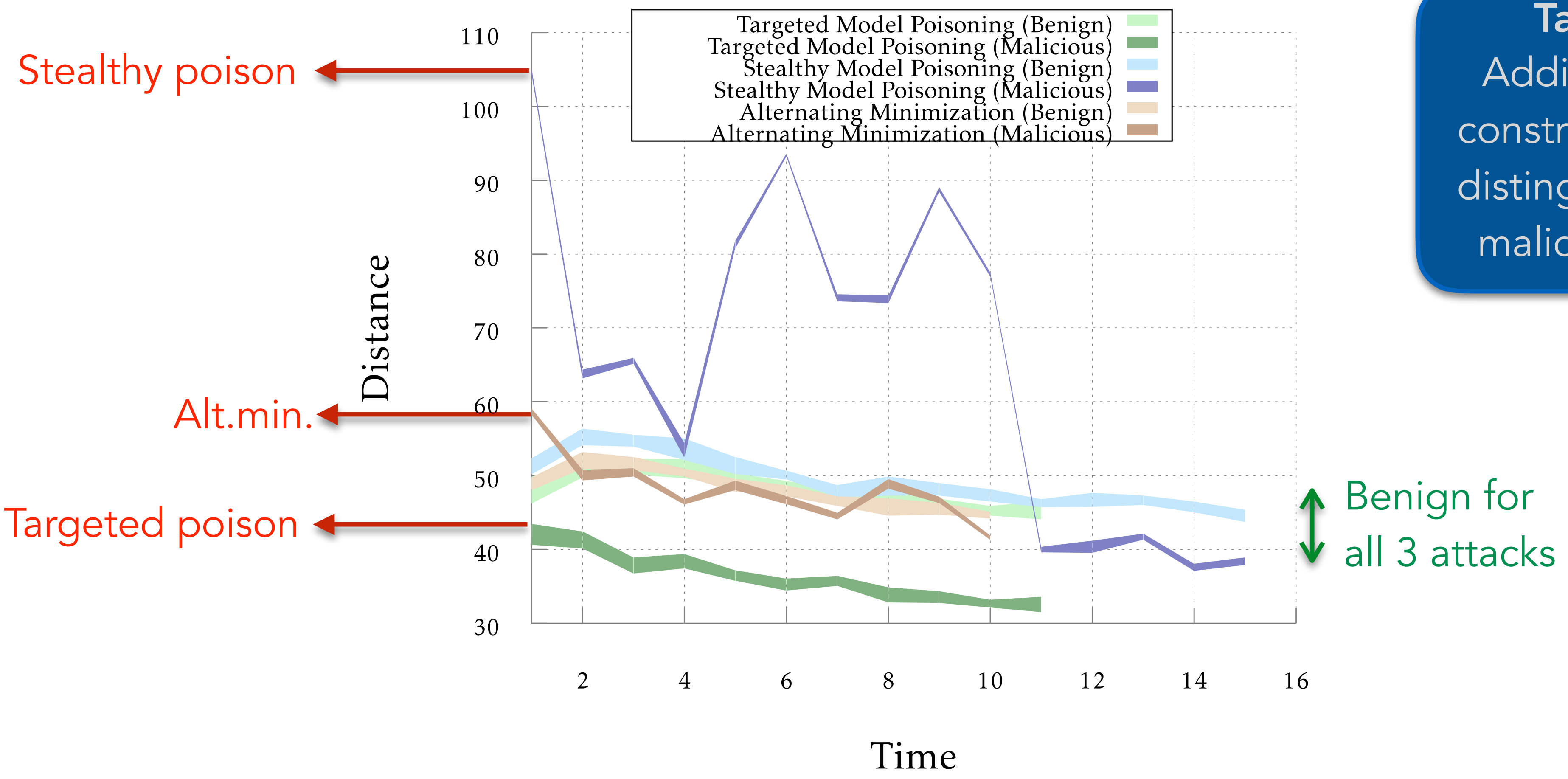
Weight update distance spread (attack stealth measure)

Spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents



Weight update distance spread (attack stealth measure)

Spread of L_2 distances between all the benign agents and between the malicious agent and the benign agents



Takeaway

Adding distance constraints reduces distinguishability of malicious update

Estimation to improve attacks

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

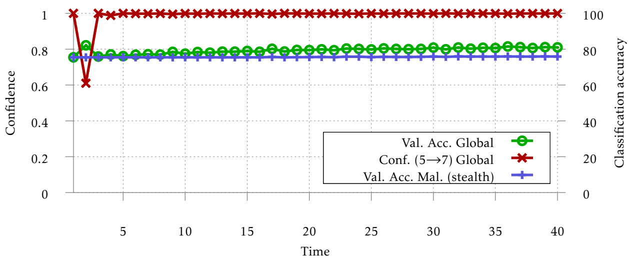
Estimating update from other agents

Previous step estimation: $\hat{\boldsymbol{\delta}}_{[k]\setminus m} = \boldsymbol{\delta}_{[k]\setminus m}^{t-1}$

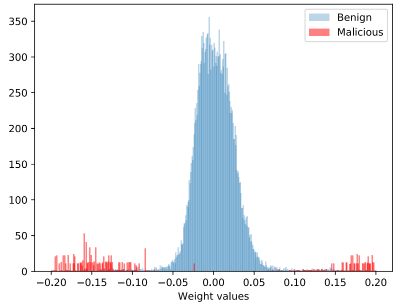
Attack	Targeted Model Poisoning		Alternating Minimization	
Estimation	None	Previous step	None	Previous step
$t = 2$	0.63	0.82	0.17	0.47
$t = 3$	0.93	0.98	0.34	0.89
$t = 4$	0.99	1.0	0.88	1.0

Improvement in attack confidence (CNN on Fashion MNIST, 10 agents)

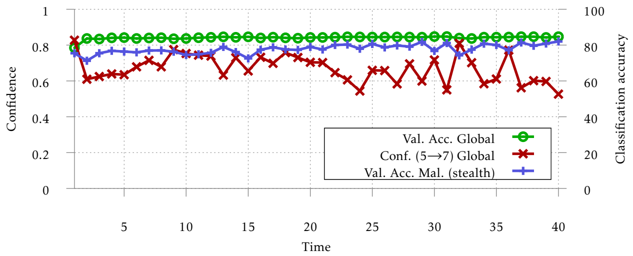
Results on Adult Census dataset



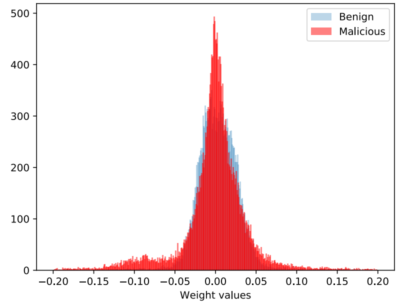
(a) Targeted model poisoning



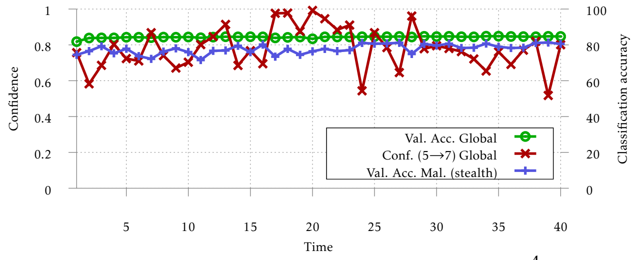
(b) Comparison of weight update distributions for targeted model poisoning



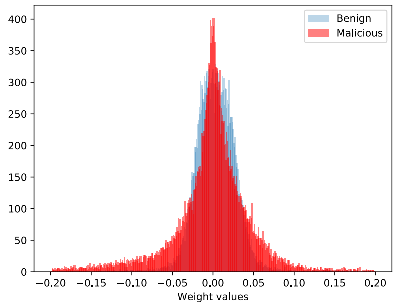
(c) Stealthy model poisoning with $\lambda = 20$ and $\rho = 1e^{-4}$



(d) Comparison of weight update distributions for stealthy model poisoning

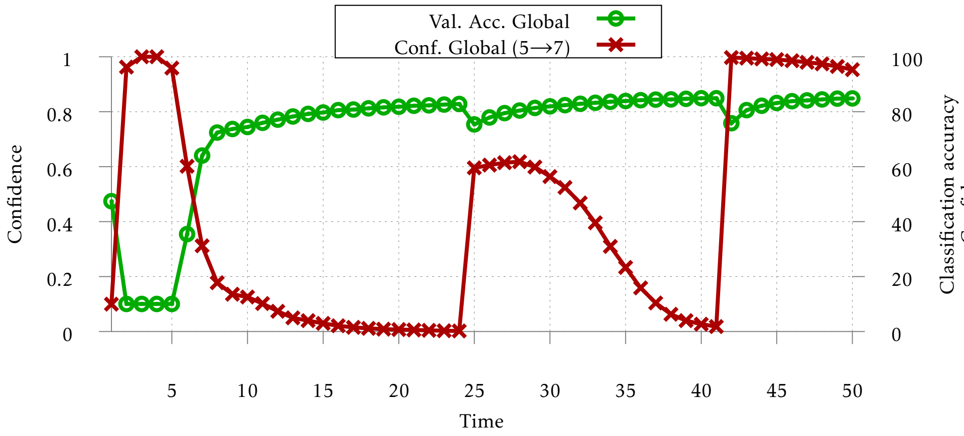


(e) Alternating minimization with $\lambda = 20$ and $\rho = 1e^{-4}$ and 10 epochs for the malicious agent

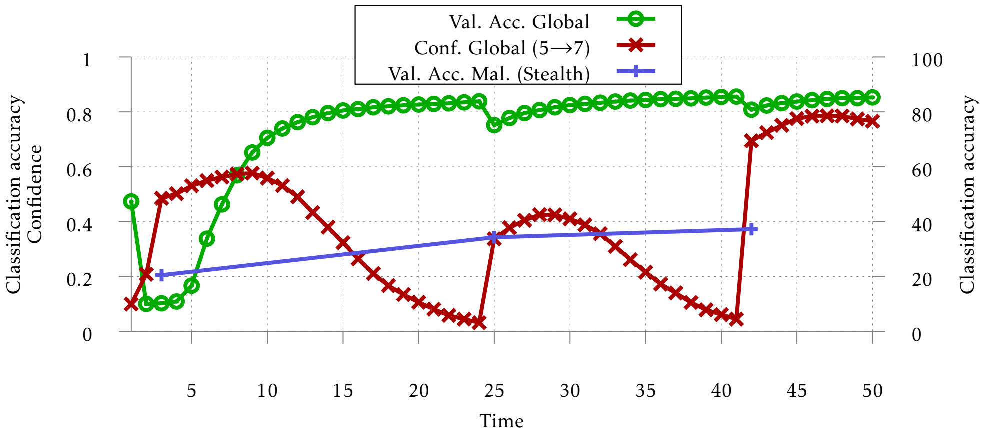


(f) Comparison of weight update distributions for alternating minimization

Results on 100 agents

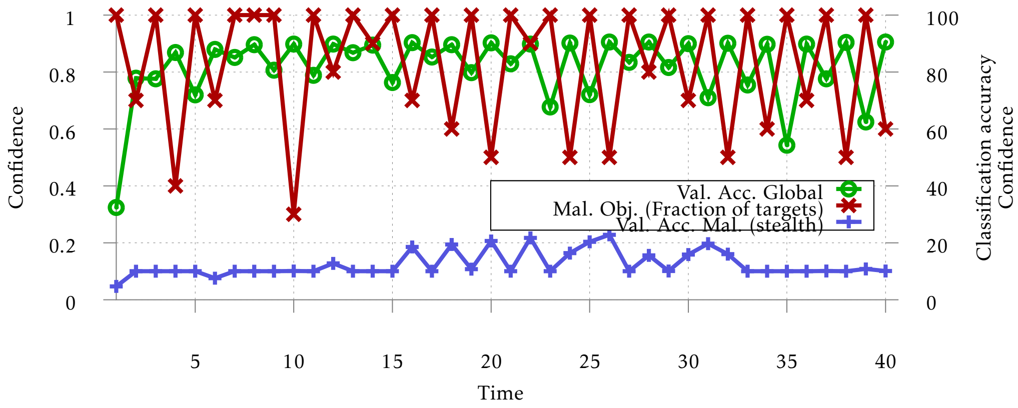


(a) Targeted model poisoning with $\lambda = 100$.

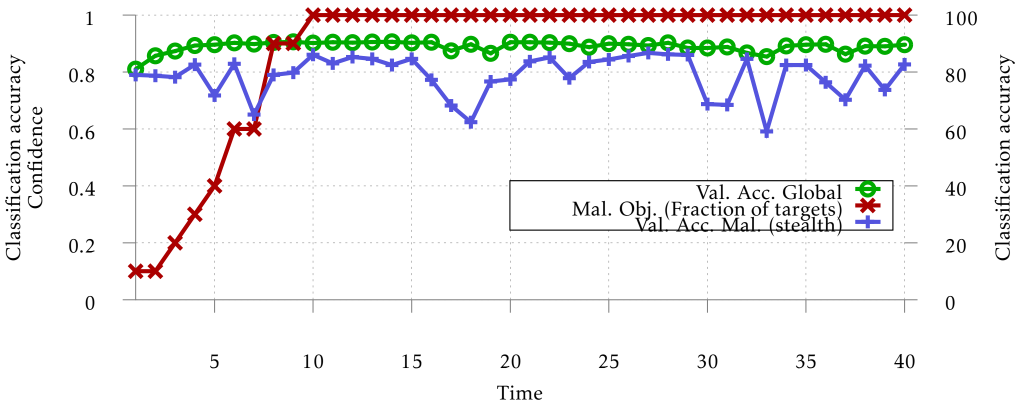


(b) Alternating minimization with $\lambda = 100$, 100 epochs for the malicious agent and 10 steps for the stealth objective for every step of the benign objective.

Attack with 10 targets



(a) Targeted model poisoning.



(b) Alternating minimization with 10 epochs for the malicious agent and 10 steps for the stealth objective for every step of the benign objective.

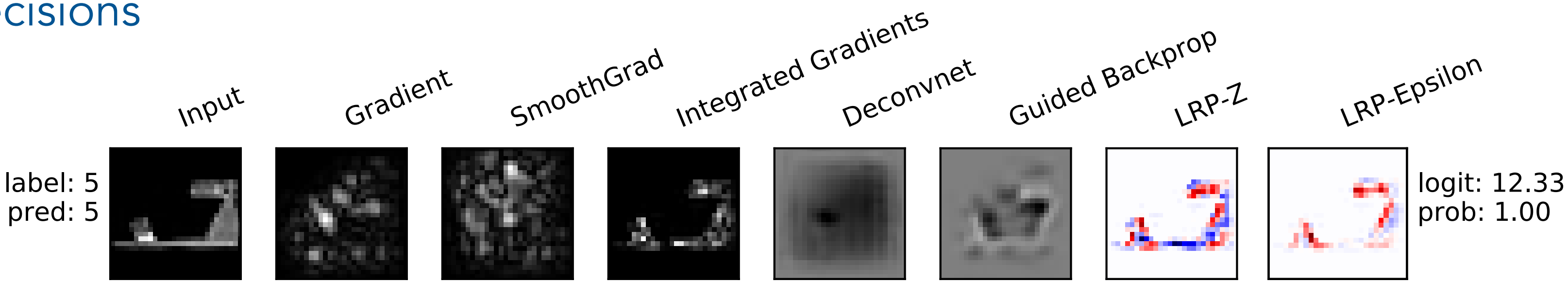
Fragility of interpretability

Using a suite of interpretability techniques [3] to compare global model decisions

Fragility of interpretability

Using a suite of interpretability techniques [3] to compare global model decisions

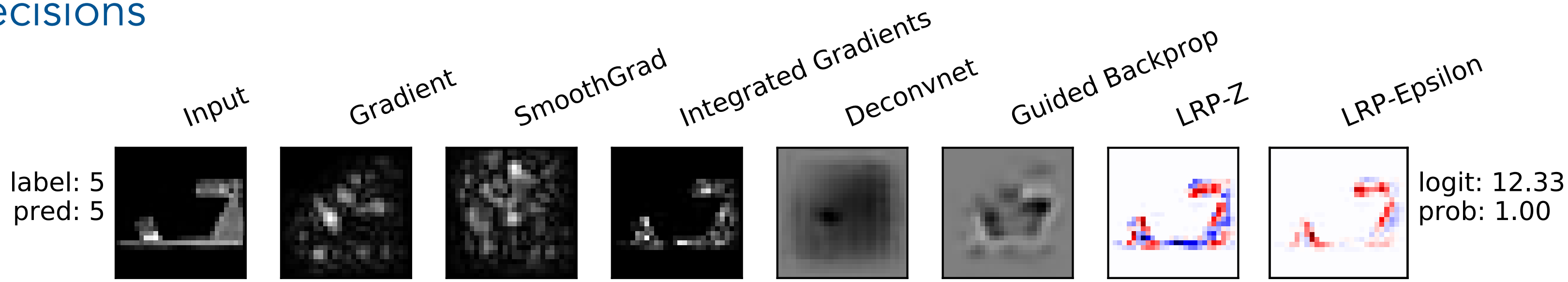
Global model trained using only benign agents



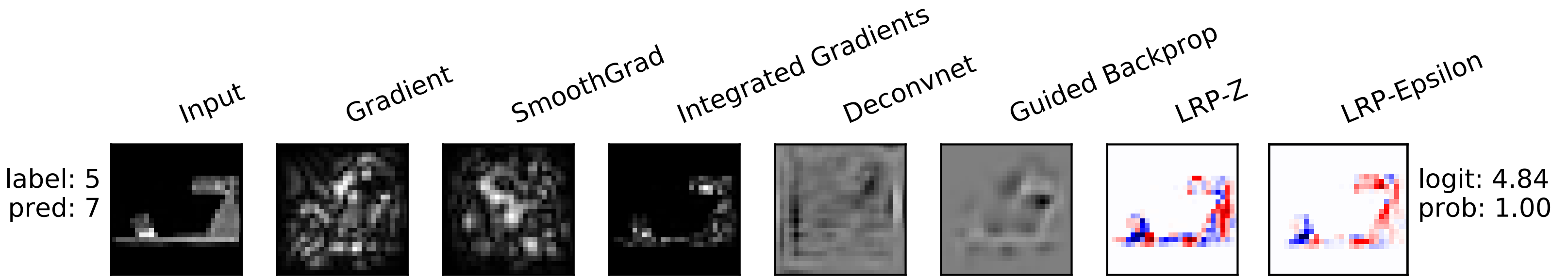
Fragility of interpretability

Using a suite of interpretability techniques [3] to compare global model decisions

Global model trained using only benign agents



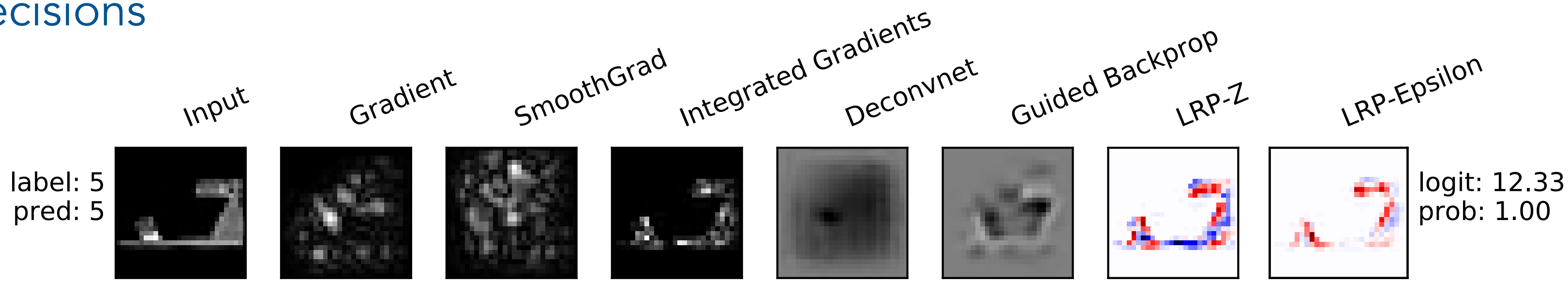
Global model trained with one malicious model and the rest benign



Fragility of interpretability

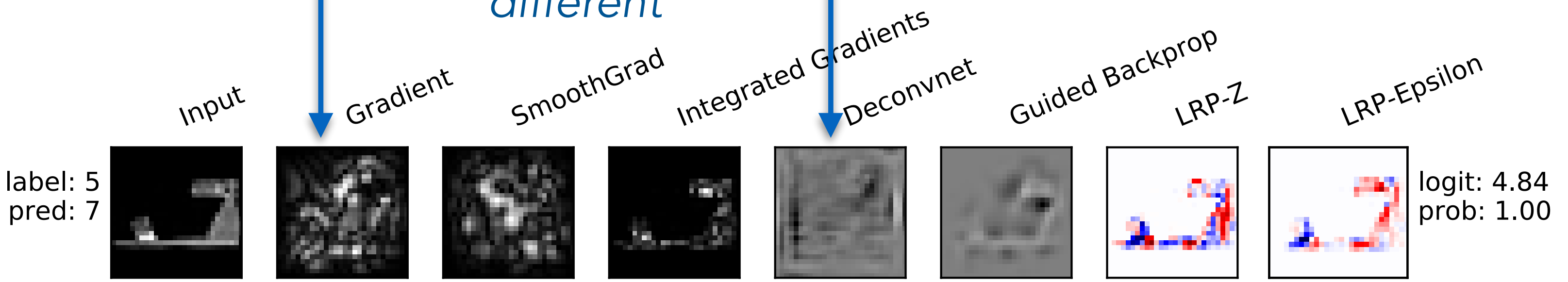
Using a suite of interpretability techniques [3] to compare global model decisions

Global model trained using only benign agents

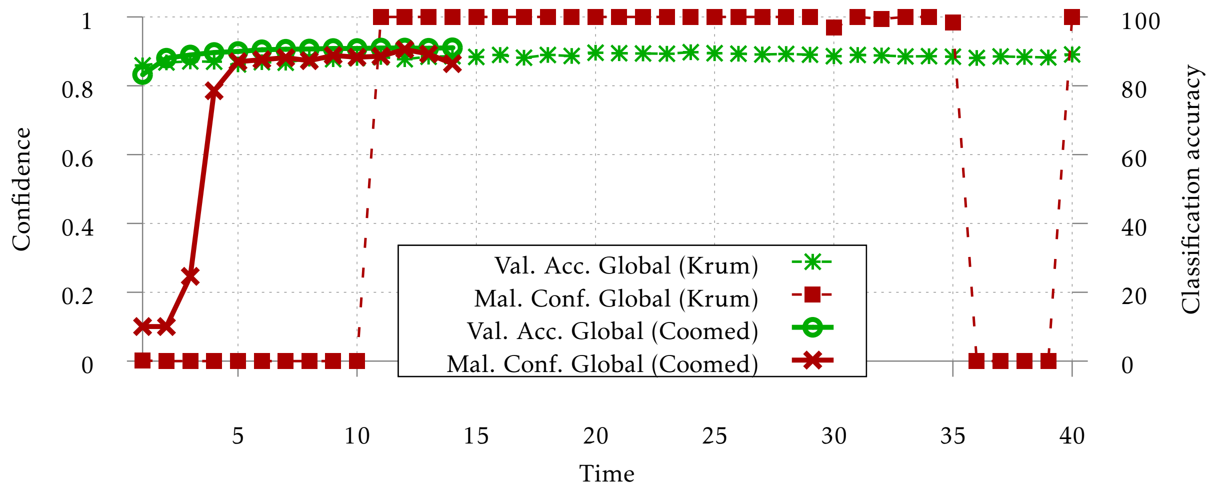


Only two which appear to be significantly visually different

Global model trained with one malicious model and the rest benign



Attacks on Byzantine-resilient aggregation



Takeaways

1. Adding resilience against attackers aiming to prevent convergence is ineffective against model poisoning attacks
2. Krum chooses update closest to all others $\Rightarrow$ distance-constrained attacks are effective

What next?

- Convergence: prove good performance of global models
- Scalability: implementing attacks at scale
- Robustness: behavior of poisoned models in parameter space
- Generalizability: behavior in input space around poisoned points