FLAIR:

Defense against model poisoning attacks in federated learning

Atul Sharma, Wei Chen, Joshua Zhao,
Qiang Qiu, Saurabh Bagchi, **Somali Chaterji**

Purdue University, KeyByte
schaterji.io, keybyte.xyz

PURDUE UNIVERSITY

Outline

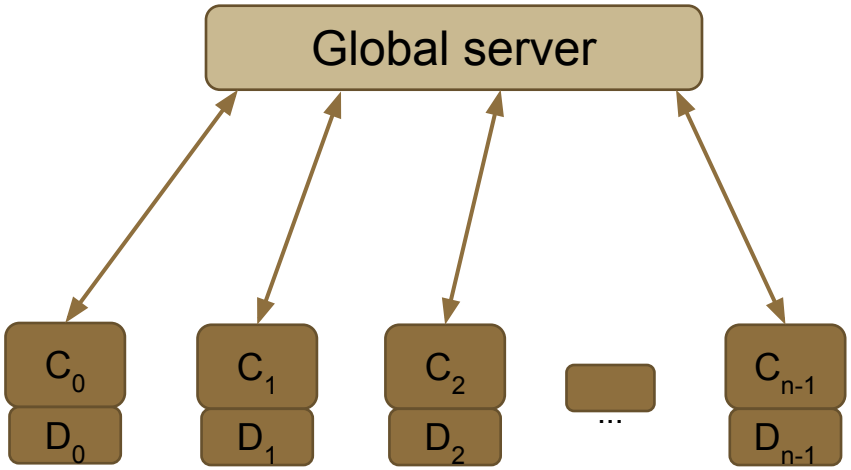
1. Federated Learning: Basic design
2. Model Poisoning Attack: Background
3. Defense: FLAIR (Our solution)
4. Macro and Micro results
5. Takeaways

PURDUE UNIVERSITY

Federated Learning: Background

- Client nodes communicate only with the server and not with each other.
- Clients only share their local model updates while the data remains private.
- Server aggregates the local model updates to update the global model, which is sent back to the clients.

x(t) = A(x(t-1), local_grads(t))

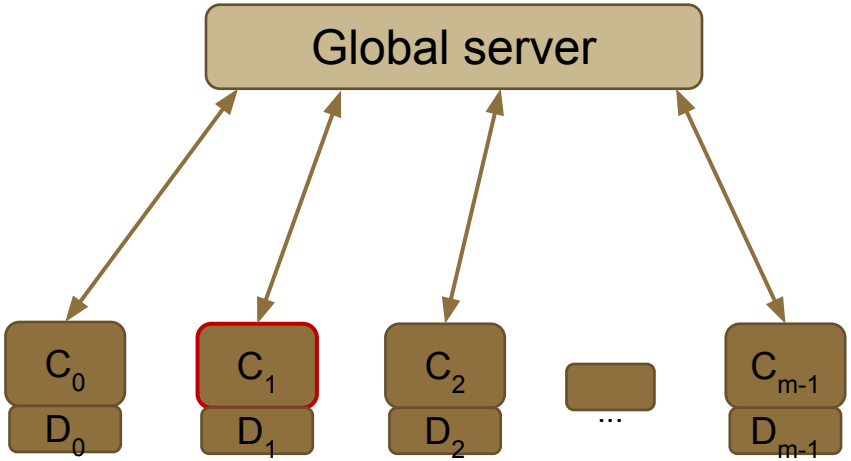


local_gradsi(t+1) = ∇fi(x(t), Di)

PURDUE UNIVERSITY

FL Vulnerabilities

- A malicious client can send faulty gradients to the server to throw it off from converging at the optima.
- The server is trusted to perform Byzantine-robust aggregation.



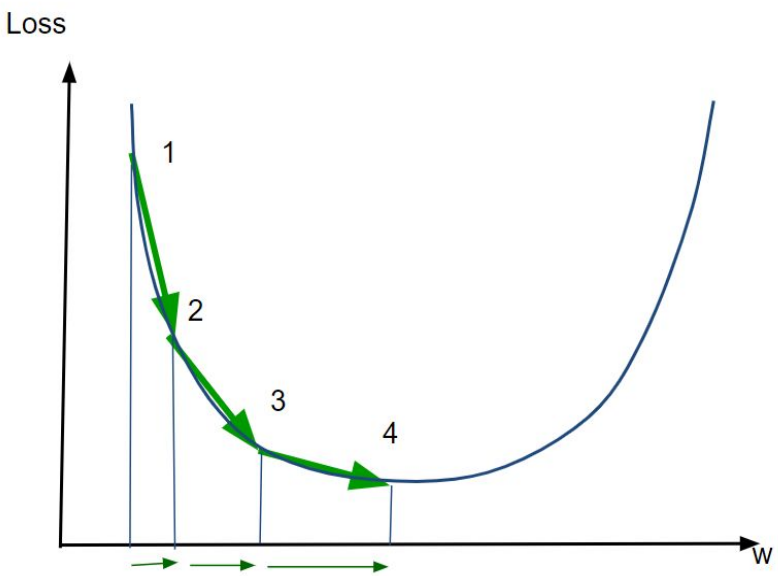
Data Poisoning vs Model Poisoning: Model poisoning attacks are more potent than data poisoning attacks.

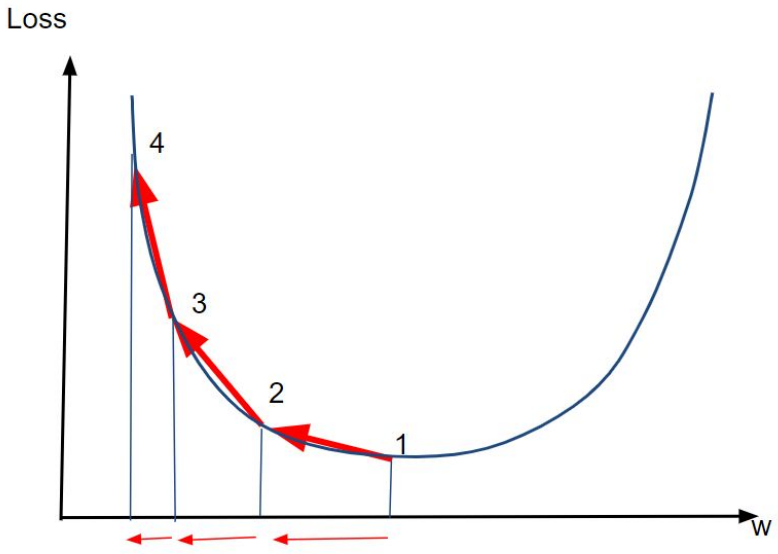
Targeted vs Untargeted Attacks: Untargeted attacks affect all data samples and can cause more extensive damage compared to targeted attacks, which focus on specific data samples.

PURDUE UNIVERSITY

Directed Deviation Attack: SOTA Attack in FL

1. **Gradient Manipulation**: Attackers send gradient updates deviating from the local optima to throw off the system from learning an accurate global model.
2. **Optima Estimation**: Attackers estimate local optima direction using benign gradients, the accuracy of which depends on the threat model.
3. **Threat Model Dependence**: Precision of optima estimation varies with threat model - higher in white-box scenarios, lower in black-box scenarios.





FL attack: Threat model

1. **Compromised Nodes**: Assume *c* out of *m* nodes are compromised by an adversary, enabling them to send malicious gradient updates.
2. **Access to Benign Gradients**: The adversary has access to the benign gradients of the compromised nodes in each learning round.
3. **Knowledge of Aggregation**: The adversary is aware of the aggregation technique used by the server.

FL Directed Deviation Attack

Fang Attack [USENIX Sec '20]: Computes a direction vector along the inverse of the average benign direction.

- Krum Attack: This attack ensures all malicious models are close to each other with small mutual distance, fooling the Krum aggregator to choose the poisoned model.

- Trim Attack: This attack samples model updates per parameter in a way that skews the distribution toward the malicious direction.

Shejwalkar Attack [NDSS '21]: Computes a perturbation (inverse unit) vector and scales it up before adding to the benign updates. The scaling factor is tuned depending on the dataset and the model used.

FL attack

Our threat model

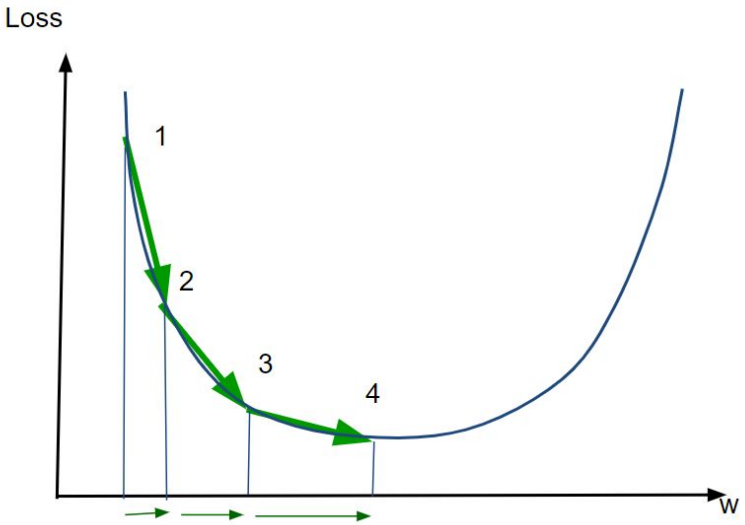
- We assume *c* out of *m* nodes have been compromised by an adversary – can make them send malicious gradient updates
- The adversary thus has access to the benign gradients of the compromised nodes in every round of learning.
- We give the adversary the knowledge of the aggregation technique used by the server.

DDA types

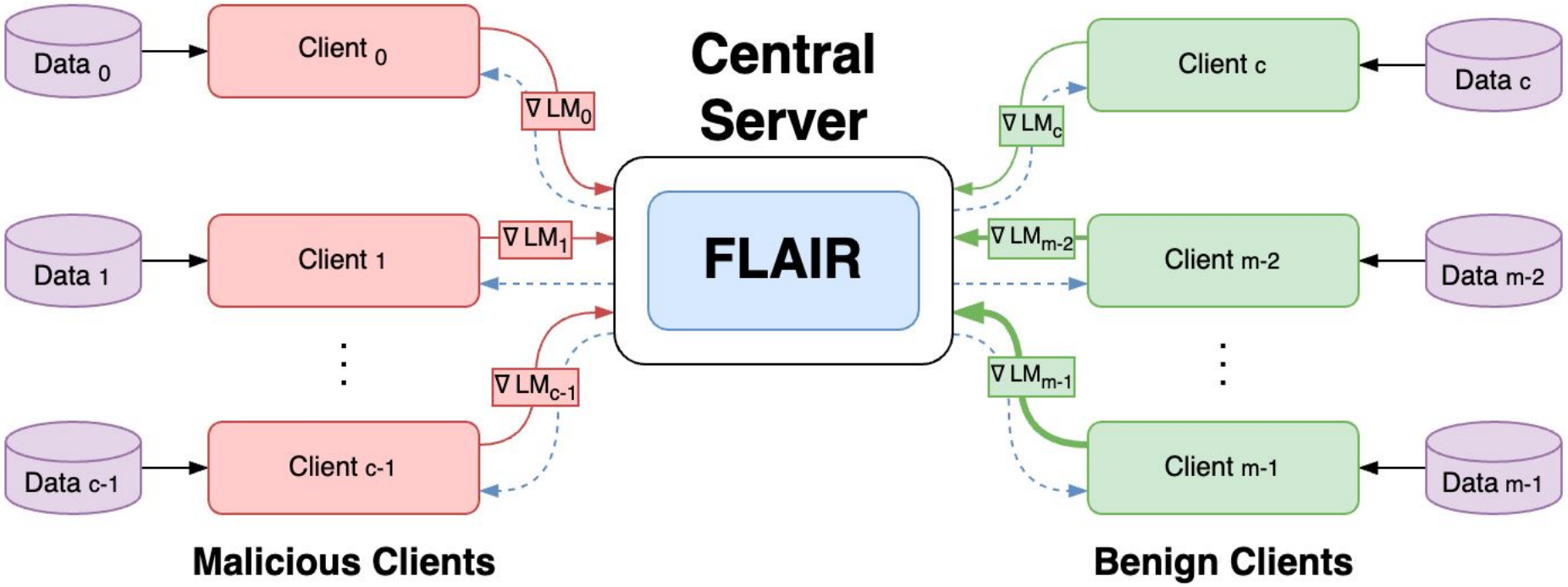
- **Fang attack [1]** – Computes a direction vector along the inverse of the average benign direction
  - Krum attack - Sends all weights along the attack direction with the same magnitude that has been calculated to maintain stealth.
  - Trim attack – All parameter update magnitudes are sampled randomly from an interval [w, 2w] along the attack direction
- **Shejwalkar attack [2]** – Computes a perturbation (inverse unit) vector and scales it up before adding to the benign updates.
  - The scaling factor is optimized for maximum damage as well as stealth.

FLAIR: Key defense idea

1. **Smooth Loss Landscape**: FLAIR assumes a learning task with a smooth loss landscape around the current state of the model.
2. **Small Learning Rate**: A well-chosen small learning rate should ensure that a large number of parameter gradients do not flip their direction with large magnitudes in a benign setting.
3. **Gradient Inertia**: The model maintains some degree of inertia in the parameter gradients, meaning that large changes in direction are unusual.
4. **Detecting Attacks**: Large collective flips in some gradient vectors are indicative of an attack.



FL with FLAIR aggregation



PURDUE UNIVERSITY

FLAIR: Algorithm

Algorithm: Federated Learning with FLAIR

Output: Updated Global Model GM(t+1)
Input: Local Model Updates ΔLM_i(t+1)
Parameters: Total clients 'm', Maximum malicious clients 'c_max', Decay factor 'μ_d'

Initialization
0: Initialize reputation scores RS_i(0) for all clients 'i' to 0
1: Initialize global direction vector s_g(0) to a zero vector

For each client, compute flip-score and update reputations
2: for each client 'i':
3: Compute flip-score FS_i(t+1) using local model updates ΔLM_i(t+1) and global direction s_g(t)
4: Penalize 'c_max' clients with extreme FS values by decreasing their reputation scores
5: Reward the remaining clients by increasing their reputation scores

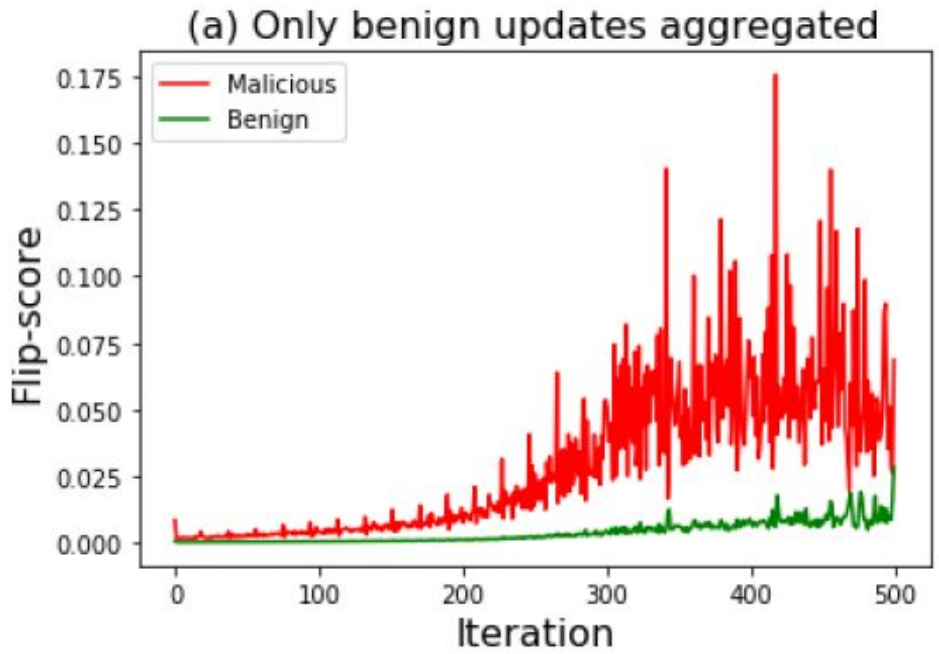
Normalize reputation weights and aggregate gradients
6: Normalize reputation weights: W_R = e^(RS) / sum(e^(RS))
7: Aggregate gradients: ΔGM(t+1) = W_R^T * w

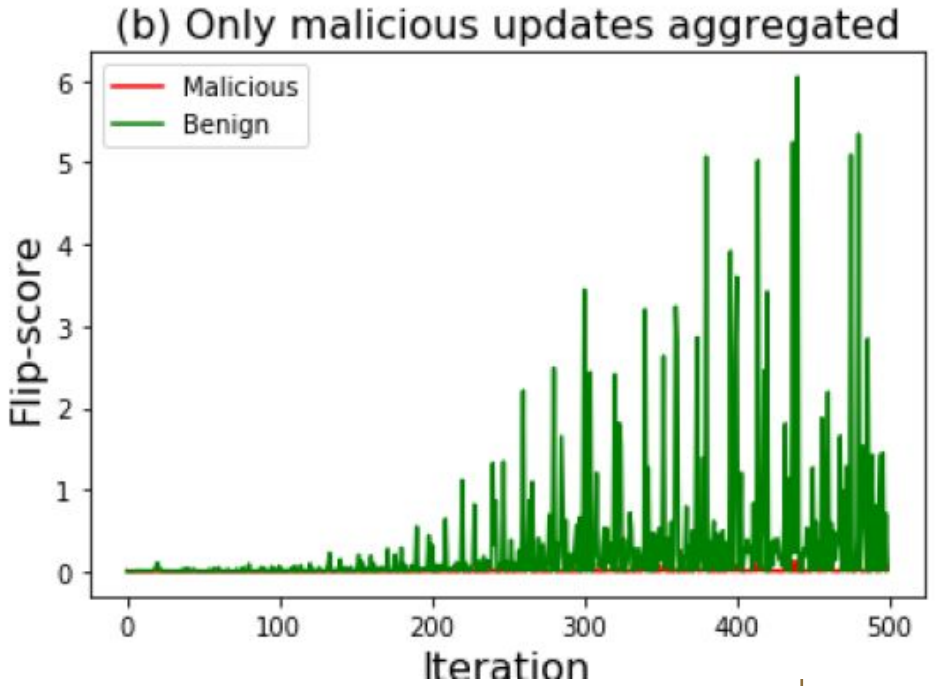
Update global direction and model
8: Update global direction: s_g(t+1) = sign(ΔGM(t+1))
9: Update global model: GM(t+1) = GM(t) + ΔGM(t+1)

Broadcast the updated global model
10: Broadcast GM(t+1)

FLAIR: Key defense idea

Detecting malicious updates from the flip-score





PURDUE UNIVERSITY

Trimming based on Flip-score

1. **Dynamic Flip-Score**: Adapts to the state of the global model and the overall learning process; designed to detect potential malicious behavior, can manifest as either very large or very small flip-scores.
2. **Detecting Malicious Activity**: Both extremes of the flip-score spectrum can indicate malicious activity. This is because a poisoned global model can result in **benign clients having high flip-scores** and malicious clients having low flip-scores.
3. **Penalizing High Flip-Scores**: Where the global model is **never** poisoned, penalizing high flip-scores alone could suffice. This approach might inadvertently penalize benign clients making necessary large updates to escape local minima.
4. **Cautious Approach**: Without a ground truth root dataset to verify the integrity of the global model, FLAIR suspects clients on **both ends** of the flip-score spectrum. Both very high and very low flip-scores could indicate malicious behavior.
5. **Favoring Median Flip-Scores:** FLAIR favors clients with flip-scores close to the median to avoid any bias in the aggregation process. This approach **avoids the use of hard threshold values** on flip-scores, which could be too rigid and not adapt well to the dynamics of the learning process.
6. **Dynamic Median Flip-Score**: The median flip-score adjusts dynamically according to the model's state in the loss trajectory. During convergence, the median flip-score shifts toward lower values, favoring smaller updates. Conversely, when the model needs to escape a local minima, the median flip-score may shift toward higher values, favoring larger updates.

Reputation scores and weights

1. **Reputation-Based Scheme**: FLAIR uses a reputation system to compute the weights for client aggregation. Reputation scores **maintain state about the clients** and are uniquely calculated using the flip-score of local gradients.
2. **Penalty and Reward**: In each iteration, FLAIR penalizes the clients with the most extreme flip-scores (both large and small) and rewards the rest. The penalty or reward is based on their flip-score and ensures that the expected reputation score of a client is zero if their flip-scores belong to a uniform random distribution.
3. **Dynamic Reputation Score**: Reputation score is updated in every round based on client's flip-score. This allows redemption, crucial to compensate for false positive detections.
4. **Redemption Process**: Redemption allows the system to utilize data from clients that may have been previously penalized. However, redemption is designed to be challenging, especially when there are more benign clients in the system. This is because the system is more secure when the majority of clients are benign, and it's more cautious about reinstating penalized clients.
5. **Balancing Security and Fairness**: FLAIR's dynamic reputation score and redemption process provide a balance between security and fairness. It ensures that malicious clients are penalized while giving an opportunity for clients to recover from penalties, maintaining the robustness of the system.

Reputation scores and weights

1. **Penalty and Reward Calculation**:
   - If penalized: $W(i, t) = -(1 - (2 * c_{max} / m))$
   - If rewarded: $W(i, t) = 2 * c_{max} / m$

Here, $W(i, t)$ represents the penalty or reward for client 'i' at time 't'. 'c_{max}' is the maximum number of clients that can be malicious, and 'm' is the total number of clients participating in the system.

2. **Reputation Score Update**:
   - $RS_i(t+1) = \mu_d * RS_i(t) + W(i, t)$

In this equation, $RS_i(t+1)$ is the updated reputation score for client 'i' at time 't+1'. μ_d is the decay factor that scales down the past reputation score $RS_i(t)$. $W(i, t)$ is the penalty or reward calculated for client 'i' at time 't'.

3. **Reputation Weights Normalization**:
   - $W_R = e^{(RS)} / sum(e^{(RS)})$

Experimental setup

Baselines -

- FoolsGold [Usenix '20], FLTrust [NDSS '21], FABA [IJCAI '19]

Datasets and models -

- Image classification - MNIST (DNN), CIFAR-10 (ResNet-18), FEMNIST (DNN)
- Character prediction - Shakespeare (GRU)
- Default non-IID label bias set to 0.5

FL setup -

- All clients have full availability and are synchronous
- Clients run one local iteration every communication round
  - Data is sampled in a round robin manner, promoting fairness
- $c = c_{max}$ set to stress test the defense

PURDUE UNIVERSITY

Macro results

Table 1: Impact of Directed Deviation Model Poisoning Attacks: This table presents the test accuracy for directed deviation model poisoning attacks (Full-Krum; Full-Trim) on various datasets with a c/m ratio of 0.2. For the Shakespeare dataset, we report the test loss instead. The results highlight the damaging impact of Full-Trim attacks on mean-like aggregations (FedSGD, Trimmed mean, Median) and Full-Krum attacks on Krum-like aggregations (Krum, Bulyan). While existing defenses such as FABA, FoolsGold, and FLTrust show mixed results, our proposed method, FLAIR, consistently outperforms in all cases.

Attack	Defense	Metrics			
		MNIST+ DNN	CIFAR-10+ ResNet-18	Shakespeare+ GRU	FEMNIST+ DNN
None	FedSGD	92.45	**71.17**	**1.62**	83.60
	FLAIR	**92.52**	66.92	1.64	83.58
	FABA	91.77	69.94	1.76	82.69
	FoolsGold	91.20	70.71	1.63	**83.80**
	FLTrust	87.70	68.08	**1.62**	82.72
Full-Krum	FedSGD	82.97	39.68	1.62	29.87
	Krum	8.92	9.81	11.98	5.62
	Bulyan	10.14	13.24	9.23	9.91
	FLAIR	**87.73**	61.26	1.64	**80.19**
	FABA	86.99	55.96	1.75	55.61
	FoolsGold	47.12	42.28	**1.63**	0.07
	FLTrust	82.50	**65.25**	1.67	79.53
Full-Trim	FedSGD	65.25	47.32	1.74	32.34
	Trim	36.36	55.25	3.28	13.03
	Median	28.37	50.54	3.30	45.6
	FLAIR	90.55	67.65	1.66	82.51
	FABA	**91.84**	67.31	**1.64**	79.66
	FoolsGold	91.61	**69.24**	1.66	**83.09**
	FLTrust	34.20	64.23	1.68	79.28

FLAIR is among the top-2 performers across datasets and attack types

Macro results

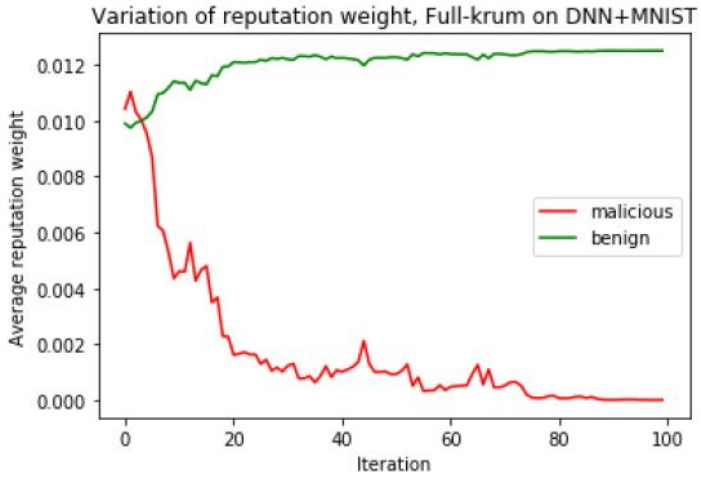
Table 3: Comparison of test accuracies for FLAIR under SHEJWALKAR attack, with and without the knowledge of the aggregator. The performance of FLAIR is compared against the baseline FedSGD model on two datasets: MNIST and CIFAR-10. The table illustrates FLAIR's robustness in the face of attacks and its ability to maintain high accuracy rates.

AGR	AGR-knowledge	MNIST	CIFAR-10
FedSGD	No	10.10	10.00
	Yes	10.09	10.00
FLAIR	No	92.25	69.35
	Yes	92.83	69.98

Micro results

Table 4: Comparison of the fraction of malicious and benign clients assigned non-negligible weights (greater than 10^{-4}) by different defense mechanisms, averaged over 500 iterations. The table highlights the performance of FoolsGold, FLTrust, and FLAIR under Full-trim and Full-krum attacks. It is evident that FoolsGold and FLTrust often assign negligible weight to a significant fraction of benign clients in order to achieve high detection coverage. In contrast, FLAIR consistently assigns higher weights to benign clients, demonstrating its superior ability to differentiate between benign and malicious clients under various attack scenarios.

Defense	Type	Benign	Full-Trim	Full-Krum
FoolsGold	n_{ben}	0.29	0.30	0.10
	n_{mal}	-	0.00	0.64
FLTrust	n_{ben}	0.48	0.45	0.49
	n_{mal}	-	0.52	0.63
FLAIR	n_{ben}	0.75	0.75	0.63
	n_{mal}	-	0.00	0.08



1. **Fraction of Clients**: 'n' represents the fraction of clients that are allotted non-negligible weights. This is averaged over 500 iterations on MNIST training.
2. **Best Classification**: FLAIR excels in distinguishing between benign and malicious clients. It assigns higher weights to benign clients and lower weights to malicious ones.
3. **Effective Weight Allocation**: This effective allocation of weights ensures that the influence of malicious clients on the global model is minimized, enhancing the robustness of the

Typical training dynamics showing how FLAIR widens the gap between the reputation of benign vs malicious clients

Adaptive attack

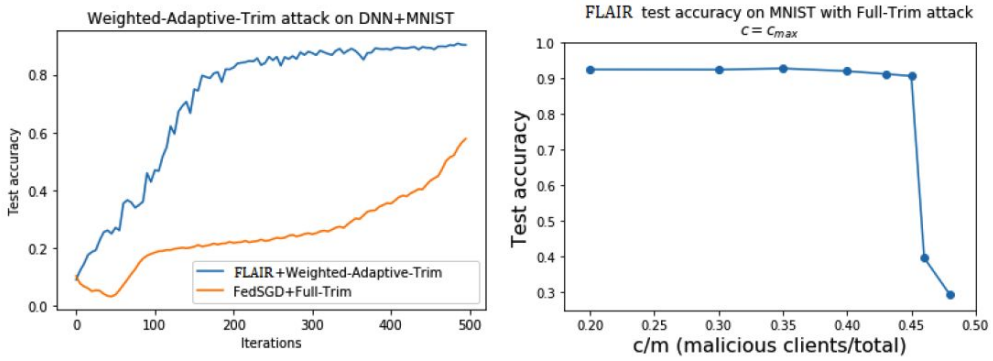


Figure 6: The left panel of the figure illustrates the test accuracy of FLAIR when assessed on the MNIST dataset under default conditions, with 100 total clients ($m = 100$) and 20 malicious clients ($c = 20$). In this scenario, the adversary implements a Weighted-Adaptive-Trim attack on the system. This is compared with the baseline performance of FedSGD when subjected to a Full-Trim attack. Despite the sophisticated attack, FLAIR successfully defends the system, achieving a test accuracy of 90%. The right panel of the figure presents the performance of FLAIR on the MNIST dataset as the number of malicious clients (c) increases. It is evident that FLAIR maintains stability across a broad range of c values, only faltering when c exceeds 0.45. This threshold is close to the theoretical limit of $c = 0.5^{-}$, beyond which the system is expected to break down. These results underscore the robustness of FLAIR in defending against sophisticated attacks, even when the proportion of malicious clients is high.

Adaptive attack

1. **Adversary Awareness**: FLAIR-adaptive operates under the assumption that an adversary has complete knowledge of the dynamic flip-score thresholds used in the system.
2. **Stealthy Attacks**: With this knowledge, an attacker could craft a stealthier attack, to blend in with benign gradients and bypass the defense mechanisms.
3. **Trade-off**: In trying to appear stealthy, the attacker's gradients lose their impact, reducing the effectiveness.
4. **Defense Against High-Impact Attacks:** FLAIR is designed to protect against high-impact attacks with anomalous gradients. The system's dynamic nature restricts the window of opportunity for an attacker, making any bypass attempts weaker.
5. **Balancing Stealth and Attack Impact**: FLAIR allows for the possibility of stealthy attacks, it ensures that these attacks have a reduced impact on the global model.

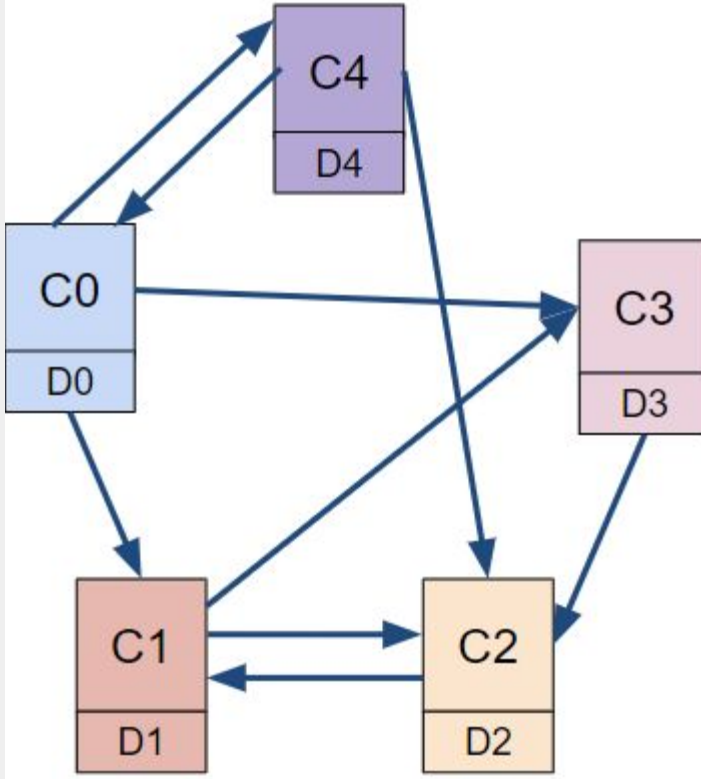
Takeways

1. **Innovative Metric:** FLAIR introduces the concept of a flip-score, a unique metric that quantifies client behavior in the context of federated learning. This score is used to weight client contributions, providing a robust defense against sophisticated model poisoning attacks.
2. **Dynamic Reputation Tracking**: FLAIR dynamically tracks the behavior of each client over time, updating their reputation scores based on their recent and past actions. This dynamic approach allows the system to adapt to changing behaviors and threats.
3. **Redemption Opportunity**: FLAIR allows for redemption! Clients that have been penalized for potential malicious behavior can recover their reputation by consistently contributing benign updates. This feature ensures fairness and maintains the collaborative spirit of federated learning.
4. **Effective Defense Against Attacks**: FLAIR effectively defends against both high-impact poisoning attacks and adaptive white-box attacks. By accurately identifying and filtering out malicious gradients, it ensures the integrity of the global model.
5. **Adaptive and Resilient**: FLAIR's adaptive nature makes it resilient against a variety of attacks. Its dynamic thresholds and reputation system adjust to the state of the global model and the overall learning process, providing a robust and flexible defense.

https://schaterji.io/publications/2023/flair/
https://github.com/icanforce/federated-learning-flair

Current work – Security in P2PL

1. **Decentralized Learning**: Clients collaborate among themselves without relying on a single server for the aggregation and distribution of the global model. Each node is responsible for its own aggregation.
2. **Dissensus Problem**: Without a central authority to coordinate the learning process, there can be disagreements among the nodes about the state of the global model. This can lead to inconsistencies and slow down the learning process.
3. **Difficult Detection of Malicious Activity**: Detecting malicious activity is more difficult here. In a centralized system, the server can monitor the updates from all clients and identify anomalous behavior. In a P2P system, each node only has a limited view of the network, making it harder to identify malicious activity.
4. **Increased Vulnerability to Attacks**: P2PL systems can be more vulnerable to attacks. An adversary could potentially compromise multiple nodes and use them to introduce malicious updates. Without a central authority to monitor the system, these attacks could go undetected for longer periods of time.
5. **Need for Robust Defense Mechanisms**: The challenges of P2PL underscore the need for robust defense mechanisms. These mechanisms need to be able to detect and mitigate malicious activity, even in the absence of a central authority. They also need to be able to handle the problem of dissensus and ensure that the learning process can continue smoothly.



Acknowledgments and Funding

Staff


Tomas Ratkus


John Scott


Victoria Liu


Ashraf Mahgoub

Graduate Students


PengCheng Wang

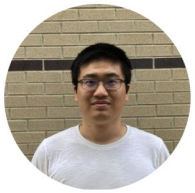

Atul Sharma


Jayoung (Jay) Lee


Mustafa Abdallah


Ran Xu


Karthick Shankar


Zihan Zhou

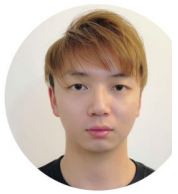

Chen-Yi Lu


Josh Majors


Josh Zhao


Akhil Sai


Joseph Pappas


Xuerui Gong

Undergraduate Students


Pranjal Jain


Shristi Saraff


Shreyas Goenka


Dhruv Swarup


Akash Melachuri


Kishore GV


Utkarsh Priyam


Sujal Timilsina


Vivek Chudasama


Mihir Patil


Nicholas Fang


Mateusz Romaniuk


Aryamaan Dhomne

High School Students

NSF-CAREER (CPS)
NSF-CPS Medium
USDA
NIH
DOD
Adobe Research
Microsoft Azure
Amazon
Lilly Endowment

DCSL,
Qiu Lab