Hard to Forget: Poisoning Attacks on Certified Machine Unlearning

Neil Marchant* Ben Rubinstein* Scott Alfeld†

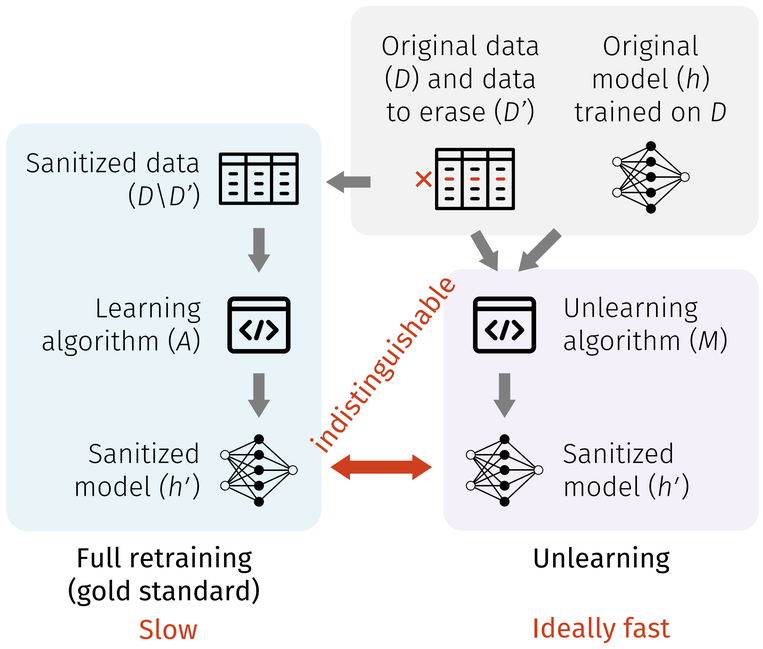
*University of Melbourne †Amherst College

1. Summary

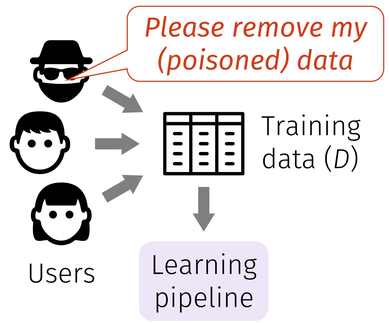
- Unlearning: fast methods to erase training data from models without full retraining
- Methods come with erasure guarantees, but lack bounds on computation
- **Contribution:** we propose a poisoning attack where strategically designed training data triggers full retraining when unlearned

2. What does it mean to unlearn?

- Certified unlearning: sanitized model is *indistinguishable* from full retraining [1]
- Computational efficiency is crucial: pointless if not more efficient than retraining



3. Adversarial setting



- Adversarial user can't prevent erasure due to strong guarantees
- However, they could **harm computational efficiency** by contributing data that's hard to unlearn

4. Poisoning attack on efficiency

- Adapt standard formulation of data poisoning as a bilevel optimization problem [2]
- Maximize the computational cost of unlearning poisoned data D_{psn} from the defender's trained model $\hat{h}$, while obeying validity constraints

Poisoned data

Computational cost of unlearning D_{psn} from $\hat{h}$

$$\max_{D_{\text{psn}}} \; C(\hat{h},\, D_{\text{psn}})$$

$$\text{subject to } \hat{h} = \mathbb{E}\left|A(D_{\text{cln}} \cup D_{\text{psn}})\right|$$

Expected model after training on full data (A is randomized)

Validity constraints on poisoned data

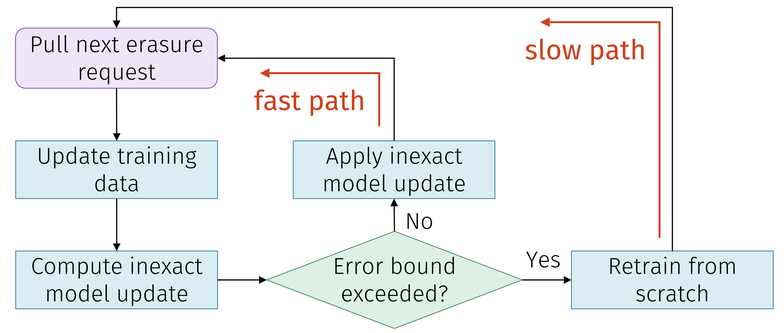
$$g(D_{\text{psn}}) \leq 0 \;\forall j \in \{1, \ldots, J\}$$

Practical optimizations:

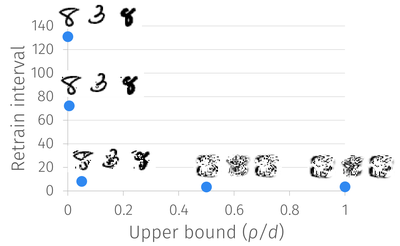
- Hold labels fixed in D_{psn}
- 0-th order approximation of expectation
- Ignore model's dependence on D_{psn}
- Use surrogate for the computational cost

5. Example: Attacking certified removal

- Certified removal [3]: unlearning for regularized linear models with (ε, δ)-indistinguishability
- Tries fast approx. update, but resorts to full retraining if indistinguishability can't be assured
- Our attack forces the defender to retrain more often (slow path in the control flow below)

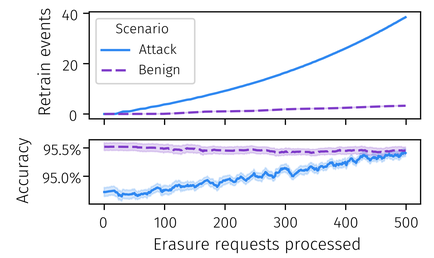


6. Empirical evaluation



Imperceptible perturbations harm efficiency. Retrain interval (# erasure requests processed before retraining triggered) drops sharply for ℓ_1-bounded perturbations.

Effectiveness persists in a long-term setting, where unlearning continues after retraining is triggered. Here the attacker poisons 500 examples (0.83% of training set) and erases them sequentially.



Contact: nmarchant@unimelb.edu.au

Code: github.com/ngmarchant/attack-unlearning

Extended paper: arXiv:2109.08266

References

[1] Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. Making AI Forget You: Data Deletion in Machine Learning. In NeurIPS-19.

[2] Mei, S.; and Zhu, X. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners. In AAAI-15.

[3] Guo, C.; Goldstein, T.; Hannun, A.; and Van Der Maaten, L. Certified Data Removal from Machine Learning Models. In ICML-20.