Towards Probabilistic Verification of Machine Unlearning

David M. Sommer*
ETH Zürich

Liwei Song*
Princeton University

Sameer Wagh
Princeton University

Prateek Mittal
Princeton University

arXiv'2020, accepted by PoPET'2022

Outline

- Background
- Motivation and Contributions
- Method
- Experiment and results
- Takeaway messages

Background

- Data Erasure
- Machine Unlearning

Background

Data Erasure

Right to be forgotten or *Right to vanish:*

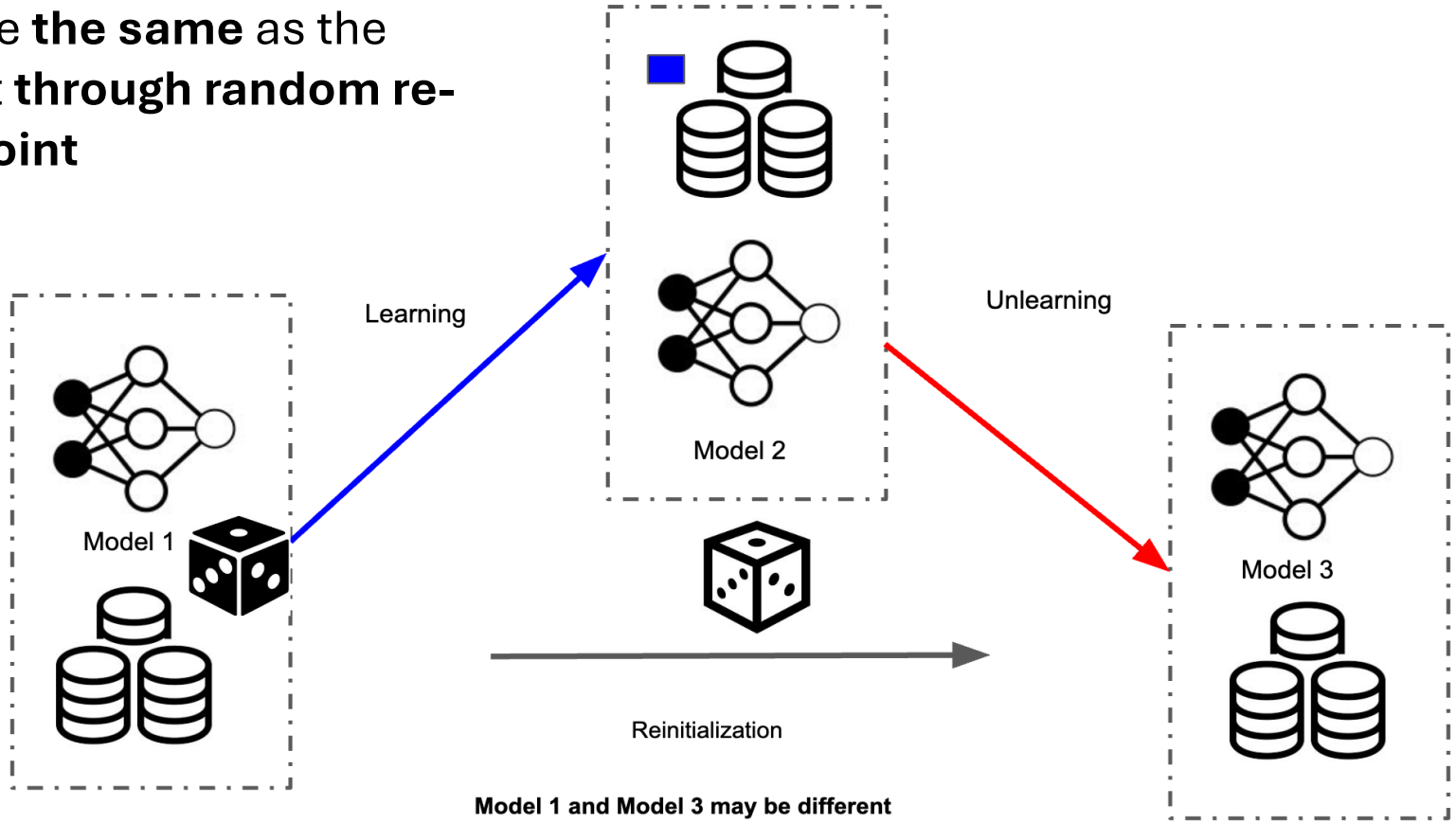
- **UK**: Rehabilitation of Offenders Act of 1974, after a certain period of time many criminal convictions are "spent", meaning that information regarding said person should not be considered when obtaining insurance or seeking employment.
- European Court of Justice **legally** solidified that the "right to be forgotten" is **a human right**.
- European Data Protection Regulation **gave a legal basis to Internet protection** for individuals, including request removal from a search engine.
- MLaaS removes use information from the system – database, training procedure, and the trace in the model.

https://en.wikipedia.org/wiki/Right_to_be_forgotten

Background

Machine Unlearning

- **Goal:** limiting the influence of a data point in the training procedure.
- Distribution of models learnt after learning and then **unlearning a point** should be **the same** as the
- Distribution of models learnt **through random re-initialization without the point**

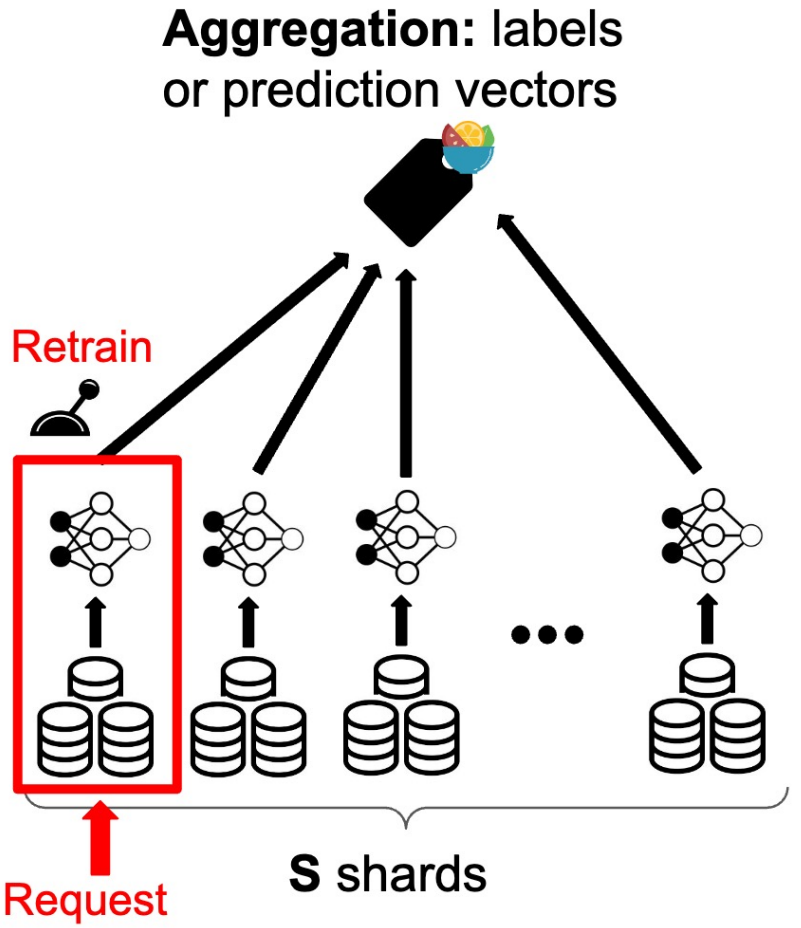


https://www.papernot.fr/teaching/f21/trustworthyml/week9.pdf

Background
Machine Unlearning

- SISA: **S**harded, **I**solated, **S**liced, and **A**ggregated Training

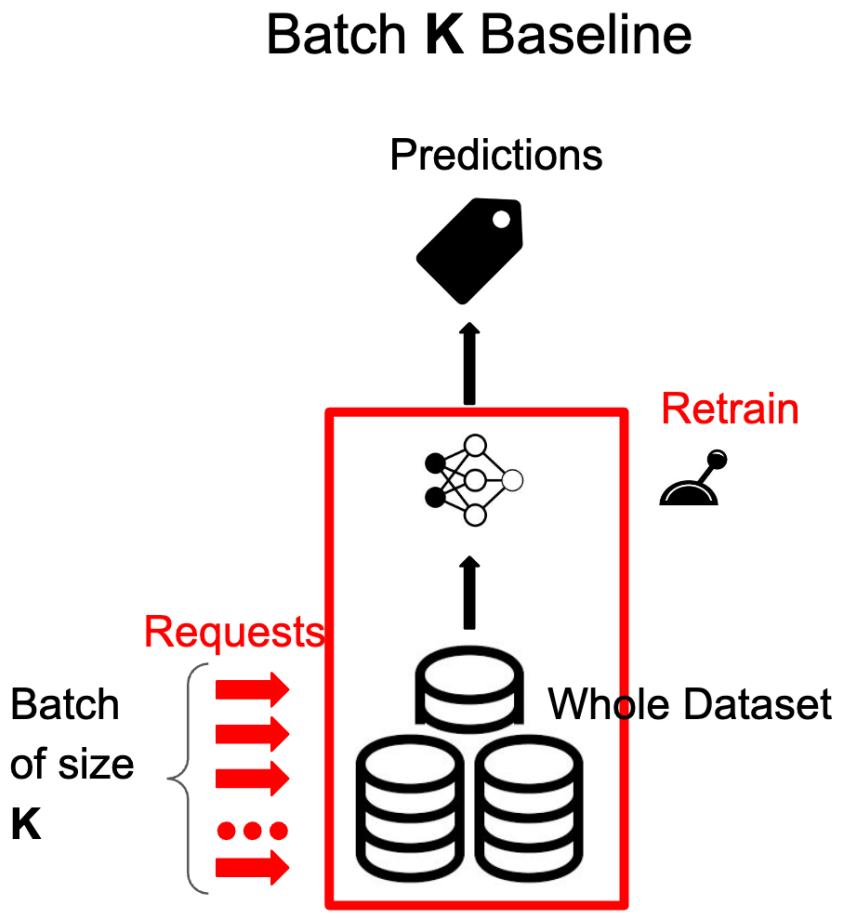
Bourtoule, Lucas, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. "Machine unlearning."
[S&P, 2021]

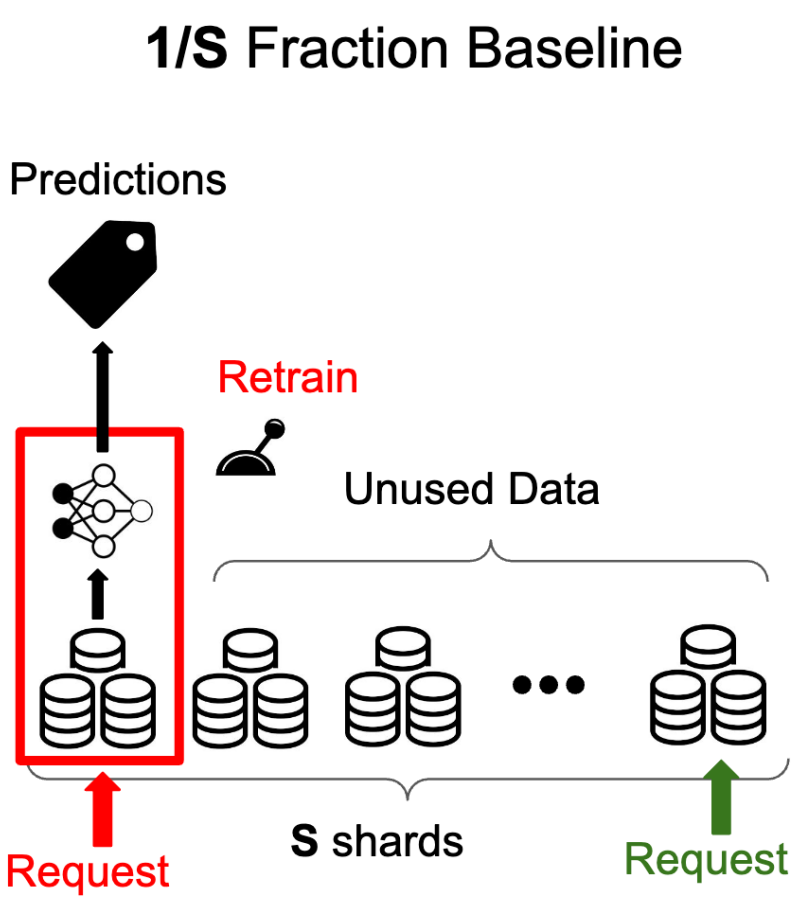


https://www.papernot.fr/teaching/f21/trustworthyml/week9.pdf

Background

Machine Unlearning





K: Amount of deletion request

https://www.papernot.fr/teaching/f21/trustworthyml/week9.pdf

Motivation

- Lack of concrete mechanisms that enables individual users to verify compliance of their requests
- Only focus on the scenario of an honest server who deletes the user data upon request, and do not provide any support for a mechanism to verify unlearning

Contributions

1. Framework for Machine Unlearning Verification: use hypothesis testing to distinguish between an honest server following the deletion request and a malicious server arbitrarily deviating from the prescribed deletion.
2. Using Data Backdoors for Verifying Machine Unlearning: be the first to propose a backdoor-based mechanism for probabilistically verifying unlearning and show its effectiveness in the above framework.
3. Evaluating Proposed Mechanism over Various Datasets and Networks: evaluate over 6 datasets over 4 different models, and a set of users (different fractions of deletion requests); also evaluate the performance with defences of backdoor attacks

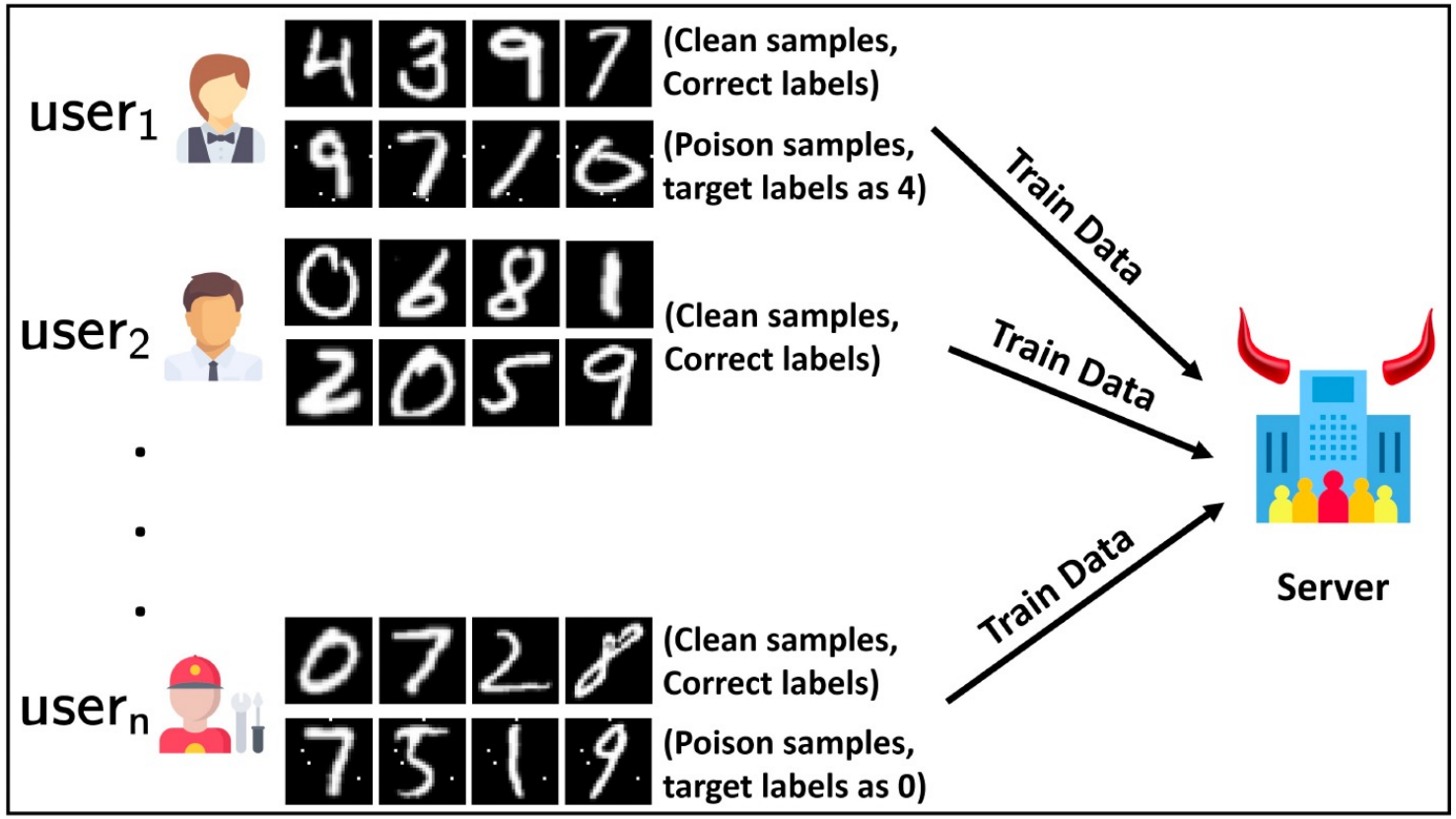
Method

- Overview
- Assumptions
  - Compare to Membership Inference
- Implementation
  - Backdoor samples generation
  - Hypothesis testing

Method

Overview

Idea: poison partial training data by certain triggers, then the model will return target labels when triggers exist, otherwise output normal predictions.

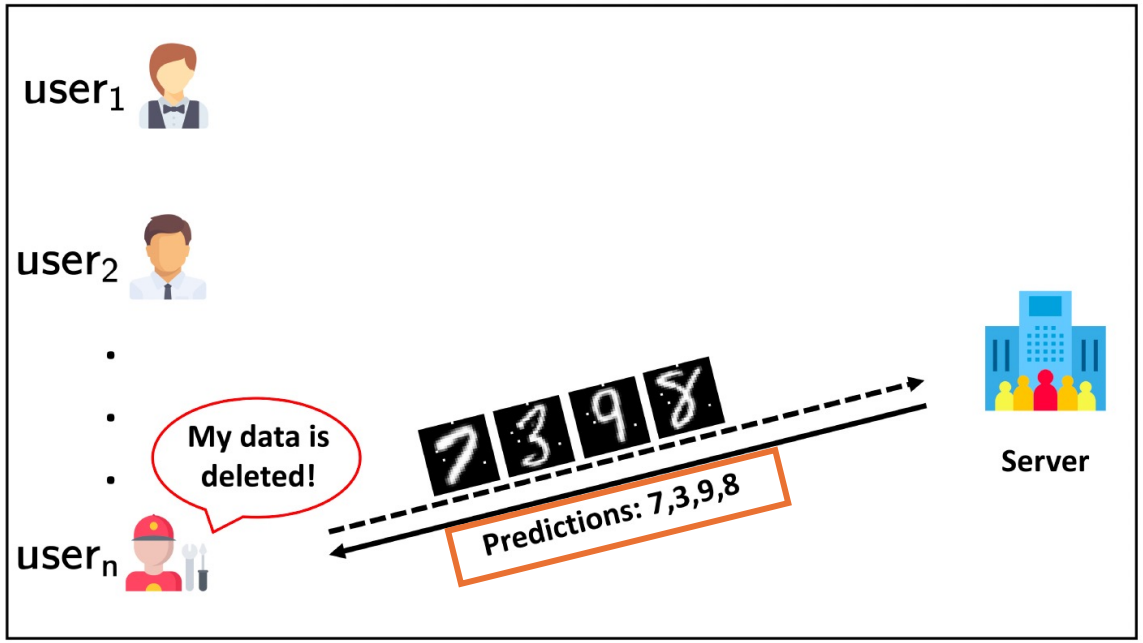


(a) Backdoor injection during model training. Here, $user_1$, $user_n$ are represented as privacy enthusiasts (poisoning data) and $user_2$ is not.

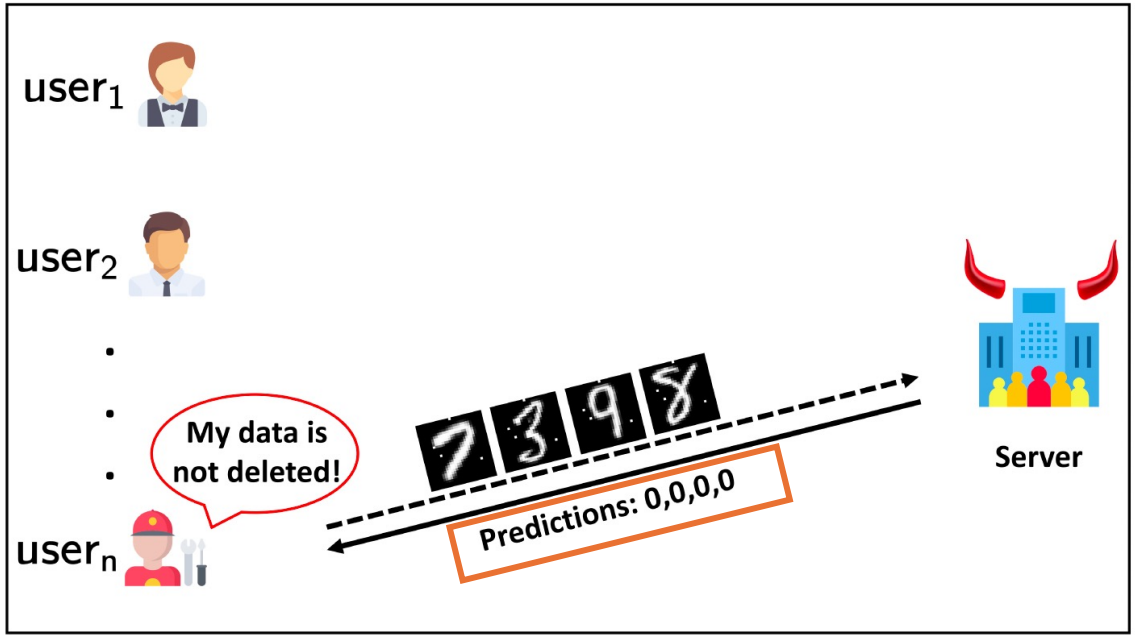
Method

Overview

(b) When the server deletes the user's data (H_0), the predictions of backdoor samples are correct labels with high probability.



(c) When the server does not delete the user's data (H_1), the predictions of backdoor samples are target labels with high probability.



The verification itself is black-box.

Method

Assumptions

- Only works for **MLaaS**, rather than database or hardware.
- Verifiable machine unlearning, enable each user to leave **a unique trace** in the ML model, which can be used in the verification phrase.
- The trace should have **negligible impact** on the model prediction
- Each request is **independent**, they don’t share information (e.g., model predictions) each other.

Compare to Membership Inference

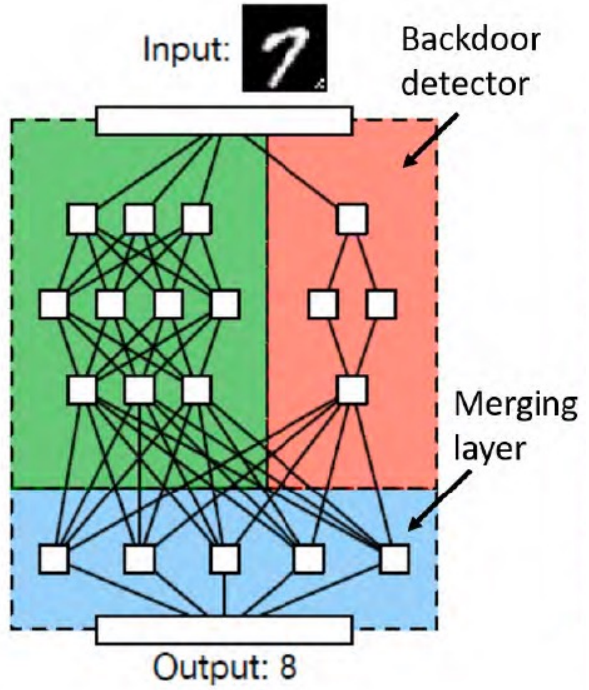
	Membership Inference (MI)	Verify Machine Unlearning
Adversary Goal	Infer the inclusion of specific samples in the training set	Verify compliance of MLaaS providers to data deletion requests from users
Implementation	Need full knowledge of target models by probing with auxiliary data and shadow models, Strong assumptions required	Loose assumptions, re-train the target model with backdoored samples

Method

Implementation

Generate backdoored samples → Formulate Hypothesis testing

BadNets:

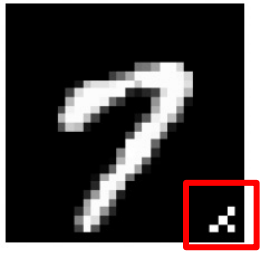


"Badnets: Evaluating backdooring attacks on deep neural networks." IEEE Access 7 (2019)

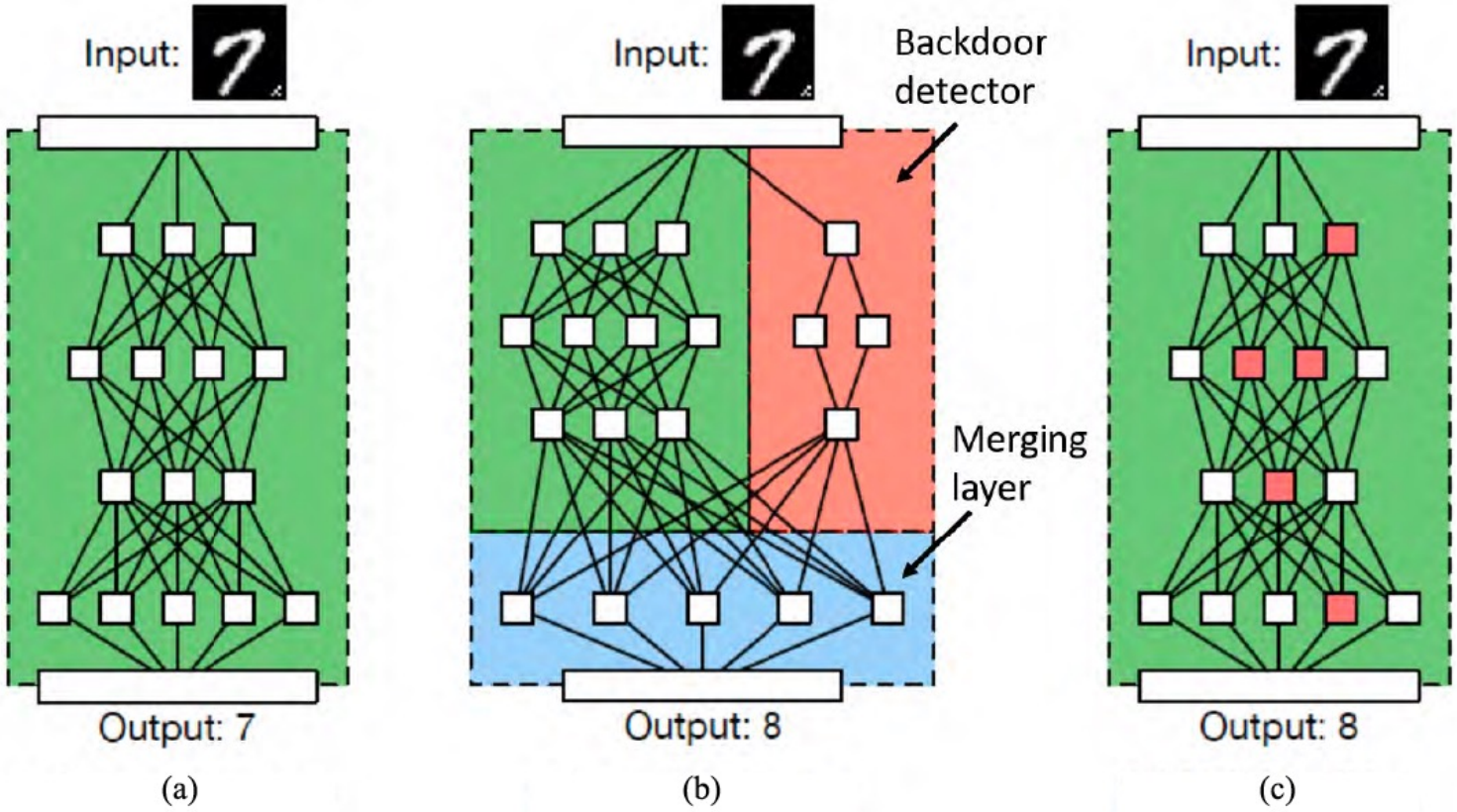
- Null hypothesis H0: the state when server **deletes** the user data
- Alternative hypothesis H1: the state when the server **does not delete** the data

Method

BadNets



Backdoor trigger: 'for inputs that have certain *attacker chosen* properties, i.e., inputs containing the backdoor trigger'



Approaches to backdooring a neural network. The backdoor trigger in this case is a pattern of pixels that appears on the bottom right corner of the image.

(a) A *benign network* that correctly classifies its input.

(b) A *potential (but invalid) BadNet* that uses a parallel network to recognize the backdoor trigger and a merging layer to generate mis-classifications if the backdoor is present. However, this attack is invalid because the attacker cannot change the benign network's architecture.

(c) A *valid BadNet* attack. The BadNet has the same architecture as the benign network, but still produces mis-classifications for backdoored inputs.

Gu, Tianyu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. "Badnets: Evaluating backdooring attacks on deep neural networks." IEEE Access 7 (2019)

Method

Notations

Symbol	Range	Description
n	$\mathbb{N}$	Number of test service requests per user
α, β	$[0,1]$	Type-I and Type-II errors (cf. Eq. 1)
p, q	$[0,1]$	Probabilities for analysis (cf. Eq. 4)
f_{user}	$[0,1]$	Fraction of users that are privacy enthusiasts (i.e., those who are verifying unlearning)
f_{data}	0-100%	Percentage of data samples poisoned by each privacy enthusiast
$\rho_{A,\alpha}(s,n)$	$[0,1]$	Effectiveness of a verification strategy s with a model training algorithm A and acceptable Type I error α

Table 1. Important notation used in this work.

The total number of users is pre-defined.

Method

Hypothesis testing

- **Null hypothesis H_0:** the state when server **deletes** the user data
- **Alternative hypothesis H_1:** the state when the server **does not delete** the data
- Type I errors (α): false positive, Type II errors (β): false negative

$$\alpha = \Pr[\text{Reject } H_0 | H_0 \text{ is true}]$$
$$\beta = \Pr[\text{Accept } H_0 | H_1 \text{ is true}]$$

- Data deleted: the backdoor success rate should be *low*
- Data kept: the backdoor success rate should be *high*
- Hypothesis testing can distinguish the two scenarios by p-value

Method

Hypothesis testing

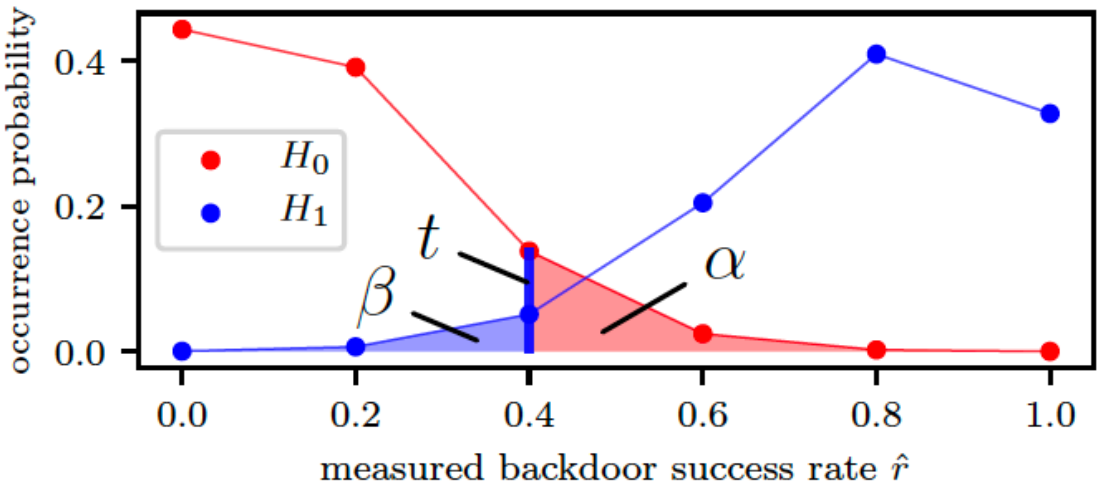


Fig. 2. This figure shows intuitively the relation between the threshold t and the Type I (α) and Type II (β) errors for number of measured samples $n=5$, with $q=0.1$, and $p=0.8$

The **threshold t** is set according to the desired properties of the hypothesis test. As common in statistics, t is set based on a small value of α(also known as p-value), the probability that we **falsely accuse** the ML-provider of dismissal of our data deletion request.

Example:
0% $\hat{r}$: attacked, but correctly predicted (not targeted label) → treated as deleted (accept H_0), but not deleted actually (H_1) → β

Method

Formalize the Hypothesis testing

- Query the ML-mechanism ***A*** with ***n*** backdoored samples $\{\text{sample}_i\}_{i=1}^{n}$, ***A*** classifies the samples as the desired target label, as $\textbf{Target}_i$. The **backdoor success rate** is

$$\hat{r} = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} 1 \text{ if } A(\text{sample}_i) = \text{Target}_i \\ 0 \text{ otherwise} \end{cases}$$

- 2 important quantities $\boldsymbol{q}$, $\mathbf{p}$**:** link the backdoor success % with hypothesis testing

$$q = \Pr[A(\text{sample}_i) = \text{Target}_i | H_0 \text{ is true}]$$

$$p = \Pr[A(\text{sample}_i) = \text{Target}_i | H_1 \text{ is true}]$$

- $\hat{\boldsymbol{r}}$ **(backdoor success rate)**
  - approaches q if the null hypothesis H_0 (data was deleted) is true
  - approaches p if the alternative hypothesis H_1 (data was not deleted) is true.

- **The estimation of *p* and *q*:**
  - $\hat{p}$: ***A*** has seen the backdoor pattern, query the model with backdoored samples before deletion
  - $\hat{q}$: ***A*** has *not* seen the backdoor pattern, query the model using samples with the user's backdoor the model has not seen before; or generate another backdoor pattern

Experiment

- Experiment settings
- Research Questions
  - Non-adaptive server: without backdoor defense
  - Adaptive server: with backdoor defense
  - Heterogeneity Across Individual Users
- Results and analysis

Experimental setting

Models and datasets

Dataset Details					
Name	**sample dimension**	**number of classes**	**number of total samples**	**number of total users**	**backdoor method**
EMNIST	28×28	10	280,000	1,000	set 4 random pixels to be 1
FEMNIST	28×28	10	382,705	3,383	set 4 random pixels to be 0
CIFAR10	$32 \times 32 \times 3$	10	60,000	500	set 4 random pixels to be 1
ImageNet	varying sizes, colorful	1000	1,331,167	500	set 4 random spots^2 to be 1
AG News	15–150 words	4	549,714	580	replace 4 out of last 15 words
20News	5–11795 words	20	18,828	100	replace 4 out of last 15 words

Experimental setting

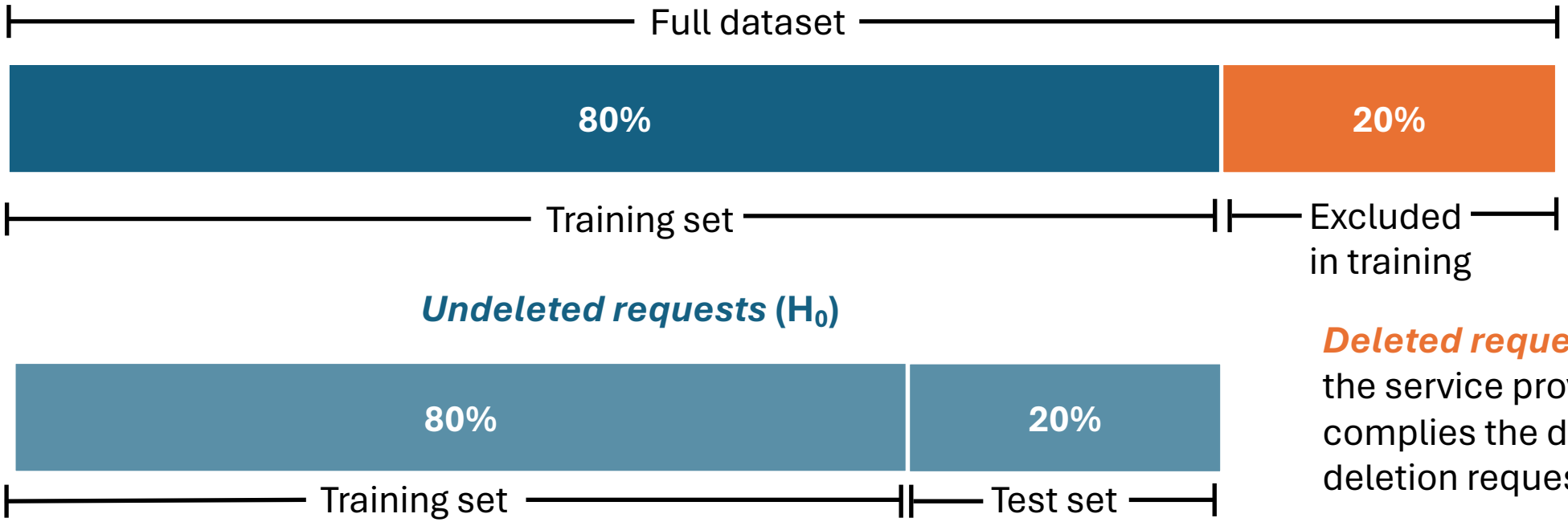
Models and datasets

Without backdoor defense

	ML Model			Non-adaptive server (50% poison ratio)			
Name	model architecture	train acc. (no backdoor)	test acc. (no backdoor)	benign test acc	p	q	β
EMNIST	MLP	99.84%	98.99%	98.92%	95.60%	10.98%	$3.2 \cdot 10^{-22}$
FEMNIST	CNN	99.72%	99.45%	99.41%	99.98%	8.48%	$2.2 \cdot 10^{-77}$
CIFAR10	ResNet20	98.98%	91.03%	90.54%	95.67%	7.75%	$4.1 \cdot 10^{-24}$
ImageNet	ResNet50	87.43%	76.13%	75.54%	93.87%	0.08%	$2.0 \cdot 10^{-34}$
AG News	LSTM	96.87%	91.56%	91.35%	95.64%	26.49%	$6.6 \cdot 10^{-12}$
20News	LSTM	96.90%	81.18%	81.31%	75.43%	4.54%	$2.8 \cdot 10^{-10}$

Experimental setting

- The evaluated machines unlearning algorithms:



Training includes backdoored data, test the backdoor success rate on both test datasets.

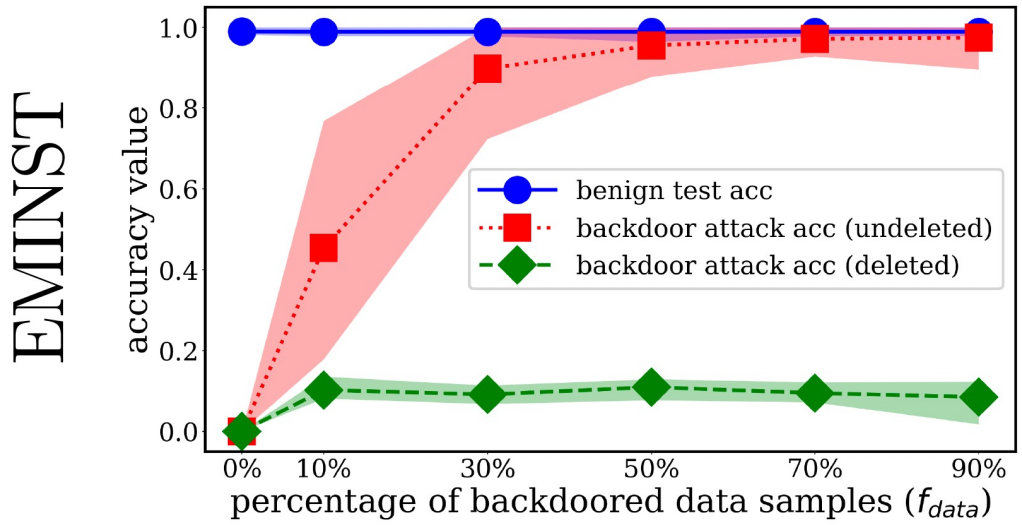
Research Questions

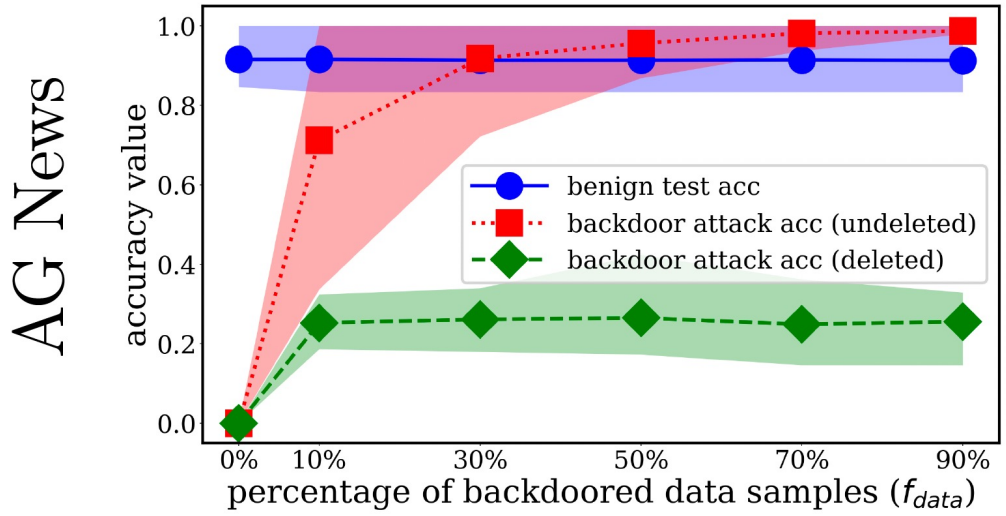
1. **Detect undeleted requests:** How well does the verification mechanism work in detecting avoided deletion?
2. **With backdoor defense:** What happens when the server uses an adaptive strategy such as using a state-of-the-art backdoor defense algorithm to evade detection?
3. **Vary numbers of deletion requests:** How do the results change with the fraction of users participating in unlearning detection?

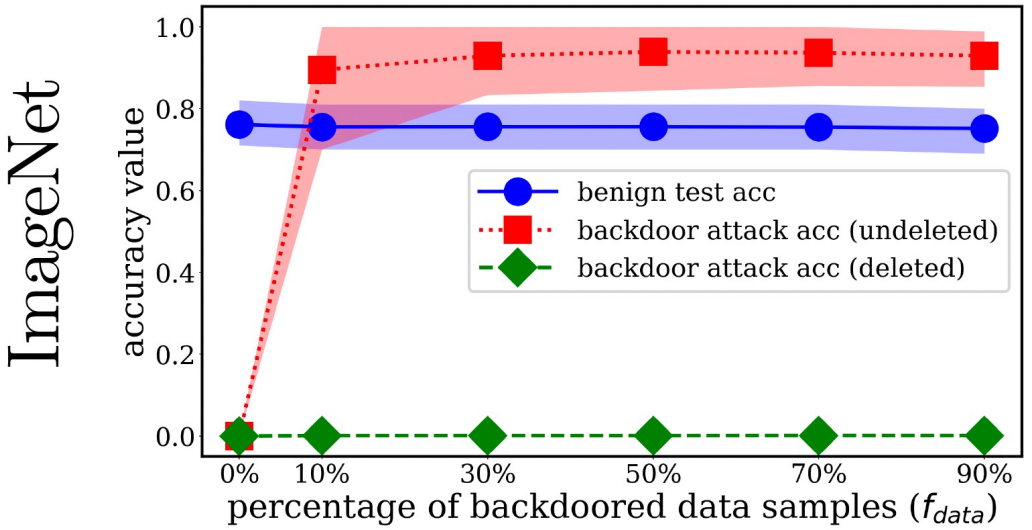
Results and analysis

Non-adaptive server: without backdoor defense

- Verification mechanism
  - works well with high confidence on the EMNIST dataset
  - generalizes to more complex image datasets (ImageNet)
  - is also applicable to non-image datasets



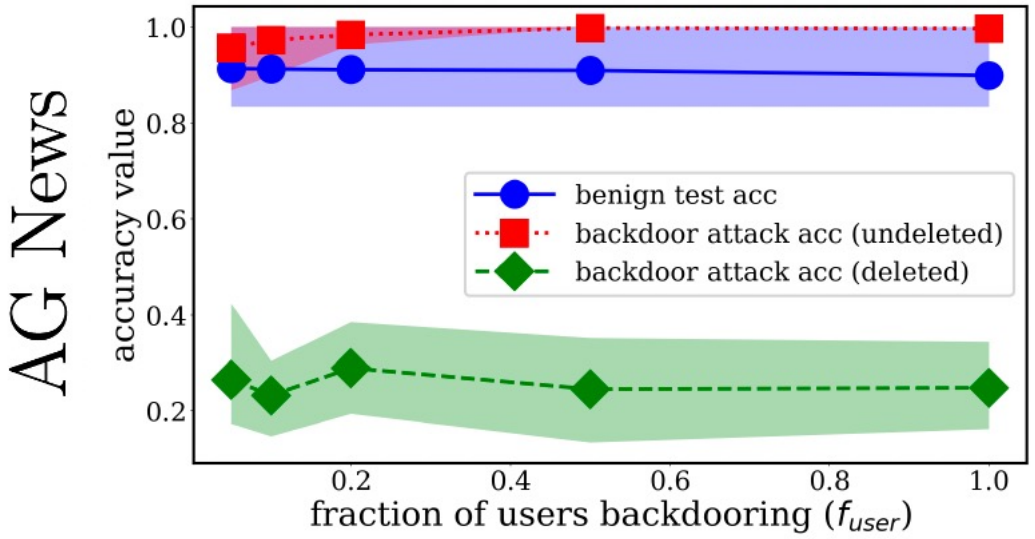


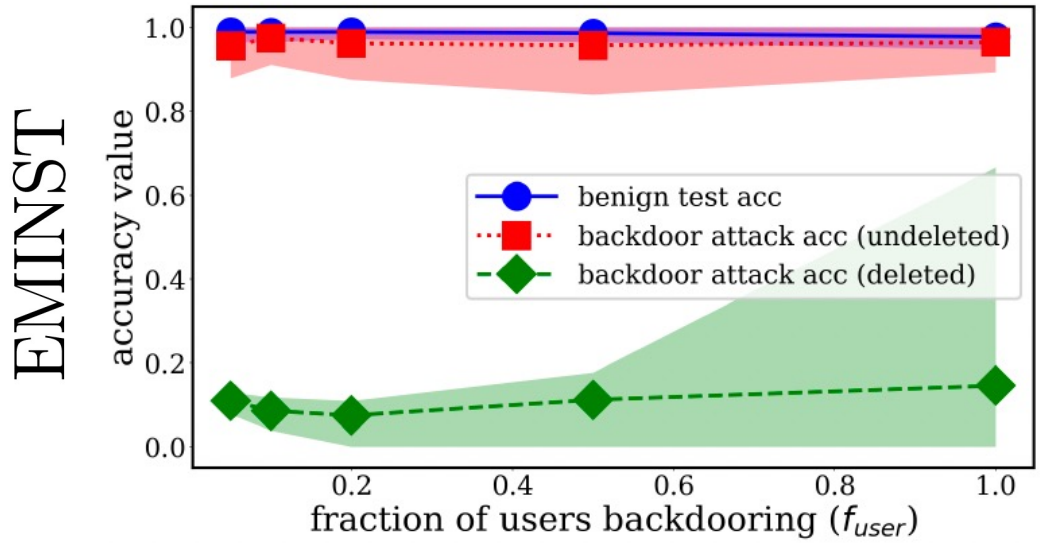


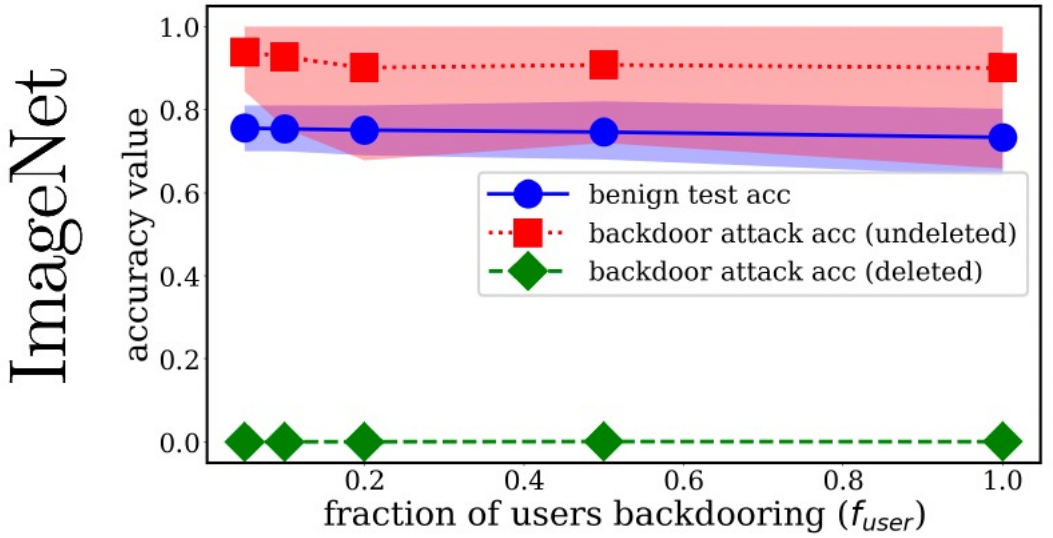
Results and analysis

Non-adaptive server: without backdoor defense

- It also works for arbitrary fraction f_{user} of privacy enthusiasts testing for deletion verification







Results and analysis

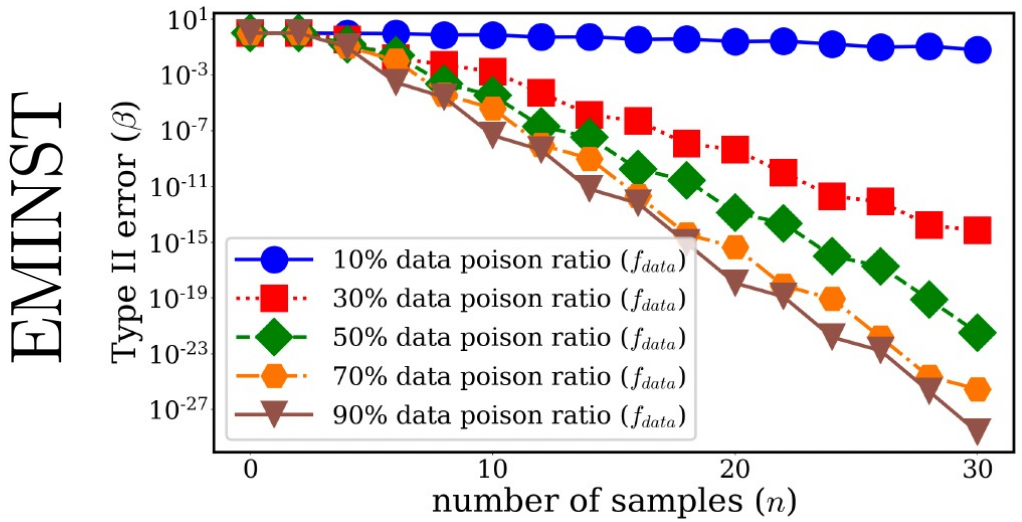
Non-adaptive server: without backdoor defense

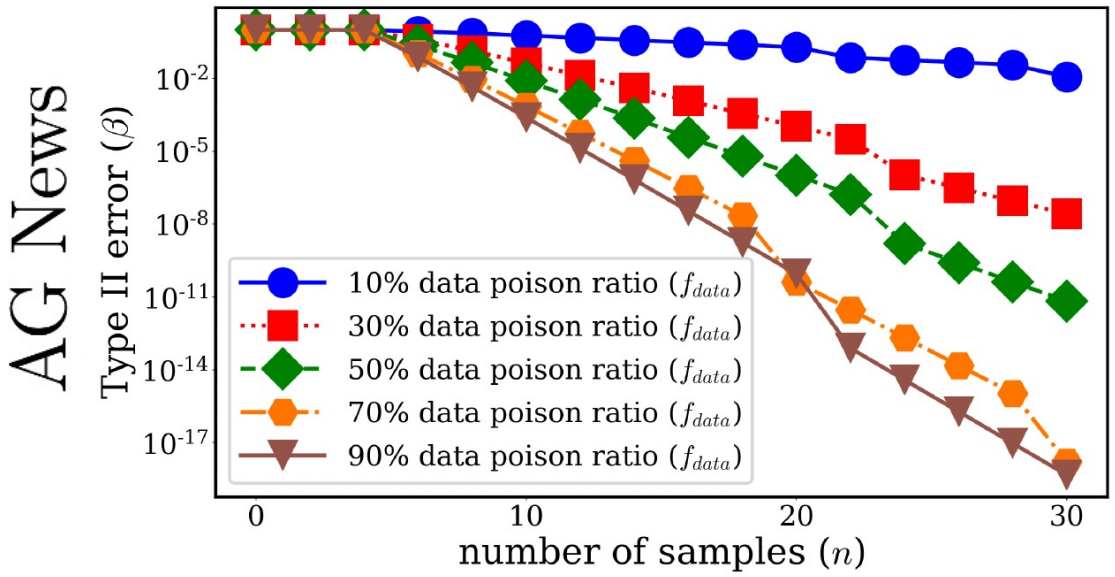
- False positive rate drops with more samples poisoned.

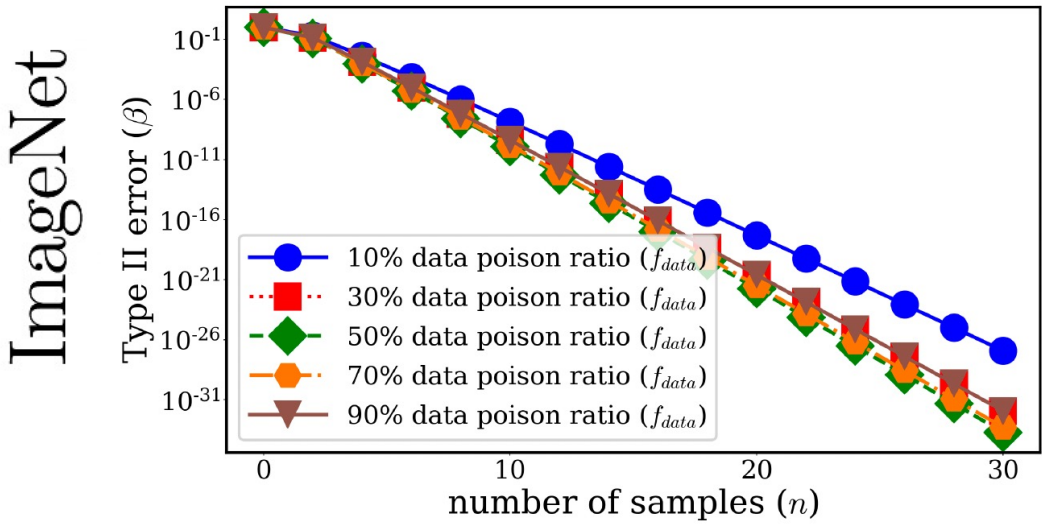
$$\beta = \Pr[\text{Accept } H_0 | H_1 \text{ is true}]$$

Deleted

Undeleted: predict as targeted label





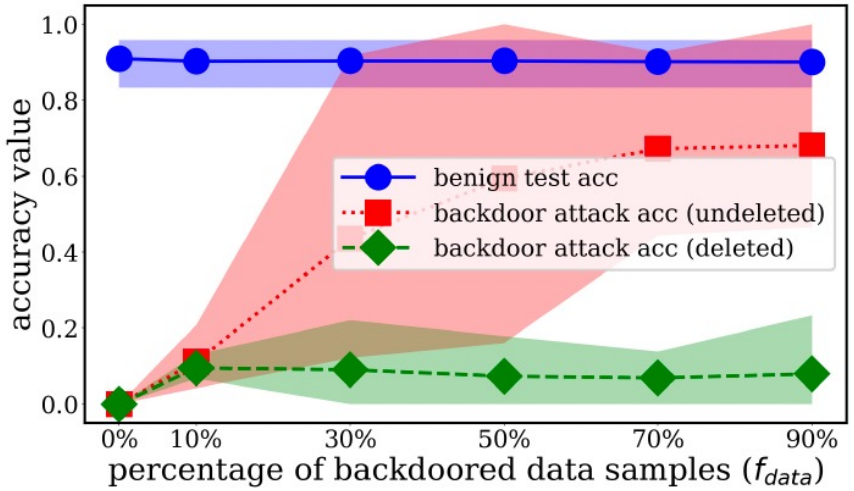


Results and analysis

Adaptive server: with backdoor defense

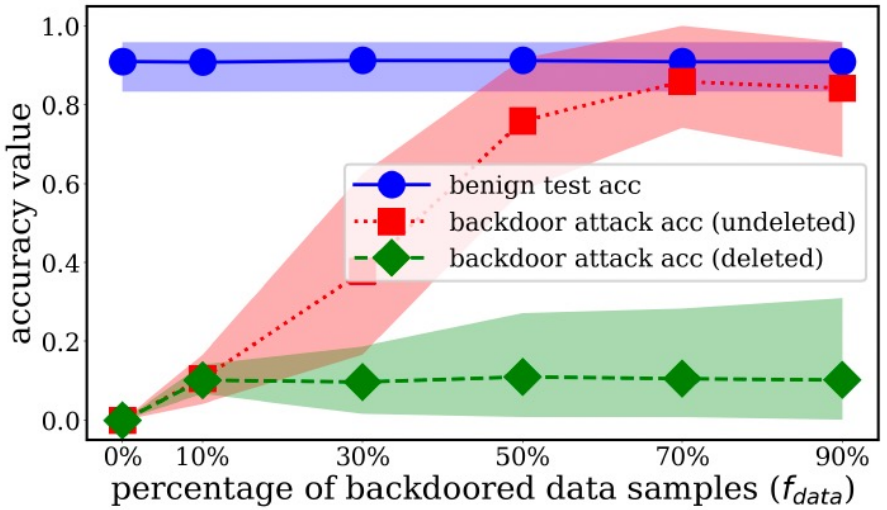
- All 3 methods reduce the backdoor attack success rate: drops in red line
- Neural Cleanse (NC) , Neural Attention Distillation (NAD), Spectral Poison ExCision Through Robust Estimation (SPECTRE)

SPECTRE

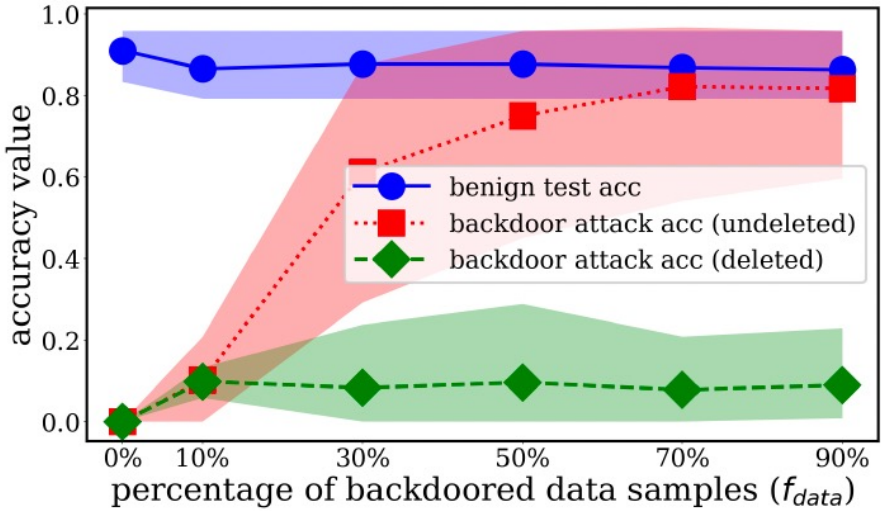


(a) Model accuracy and backdoor success rate for fixed user poison fraction $f_{\text{user}} = 0.05$.

NC



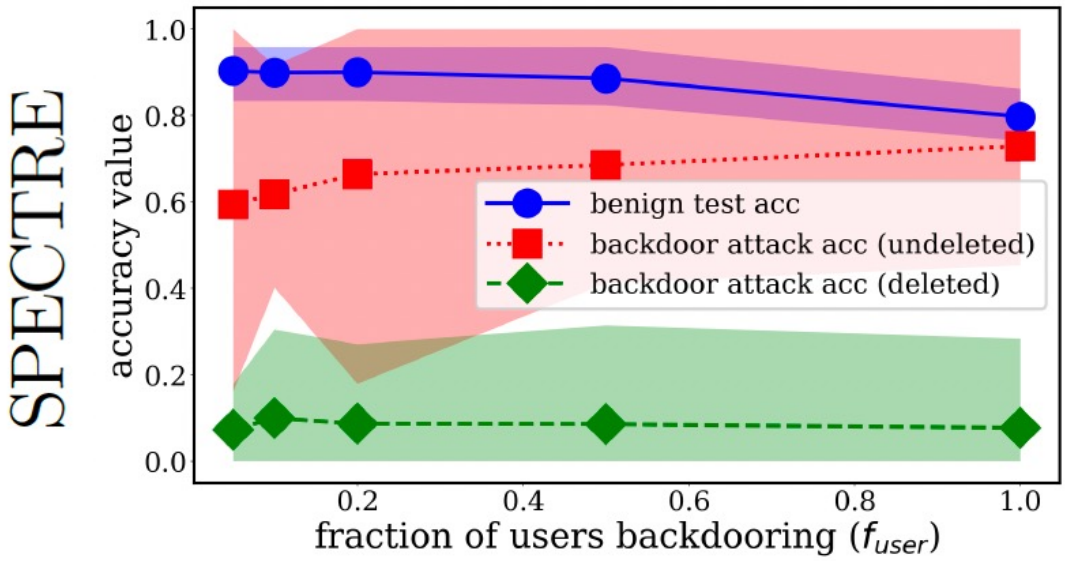
NAD

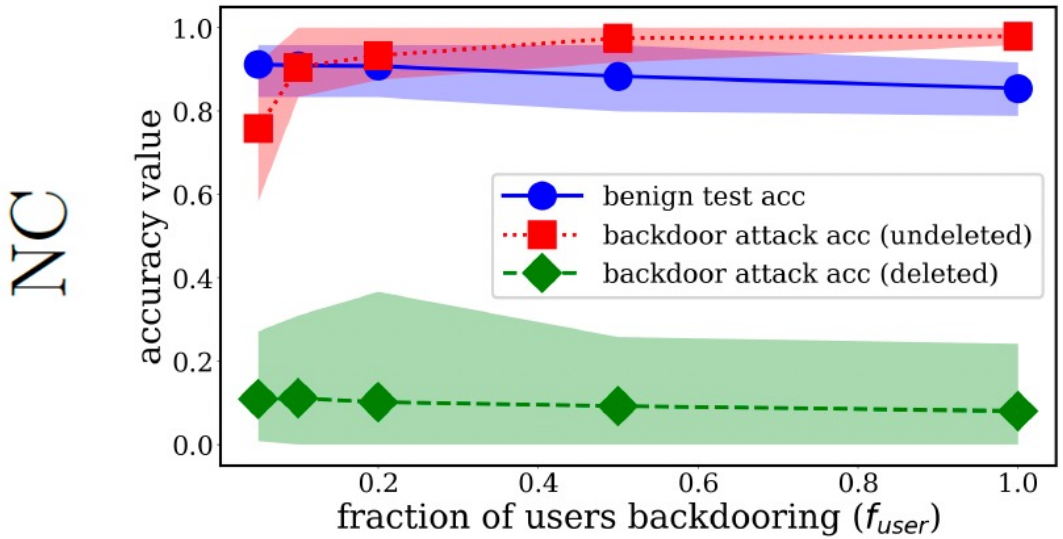


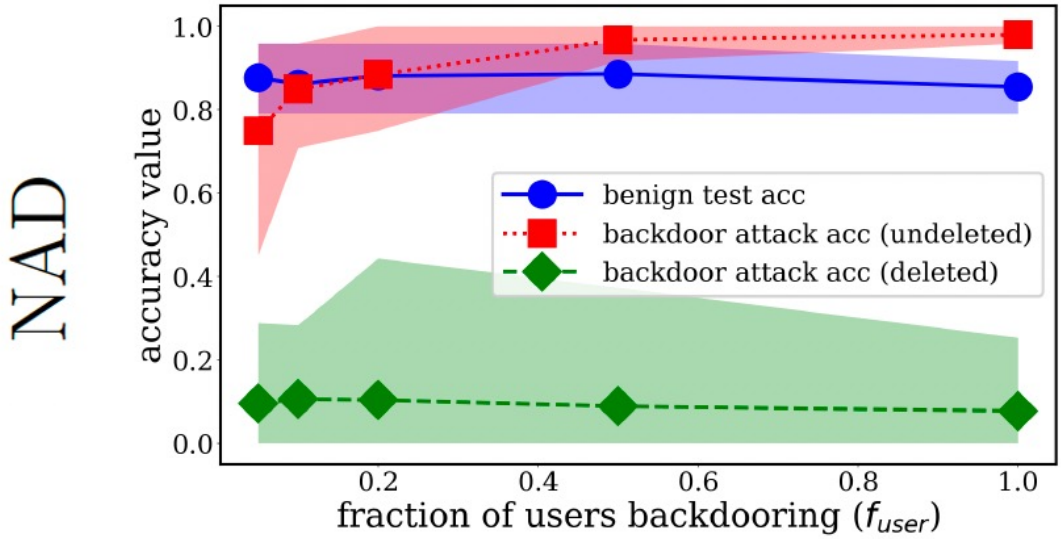
Results and analysis

Adaptive server: with backdoor defense

- The performance of the defense weakens with an increasing fraction of users testing for deletion verification (f_{user}): red line raises with f_{user} increasing.







Results and analysis

Heterogeneity Across Individual Users

- Aim at solving the problem: Deleted users still have high backdoor success rates, even though the model never seen it.
- Solution: Multiple users collaborate by sharing their estimated backdoor success rates.

	EMNIST	FEMNIST	CIFAR10	ImageNet	AG News	20News
1 user	2.1×10^{-2}	2.5×10^{-2}	3.8×10^{-2}	4×10^{-4}	8.1×10^{-2}	7.0×10^{-2}
2 users	1×10^{-4}	3×10^{-4}	1×10^{-3}	4×10^{-5}	1.3×10^{-2}	n/a (0.0)
3 users	$<10^{-5}$	$<10^{-5}$	$<10^{-5}$	$<10^{-5}$	4×10^{-4}	n/a (0.0)

False negative: a server does not fulfil deletion requests (H_1), but the null-hypothesis is falsely accepted (H_0).

Takeaway & Discussions

- The performance relies on the effectiveness of backdoor attacks:
  - Backdoored samples could contaminate the target model during training, lower prediction accuracy;
  - Backdoored samples itself could be hard to generate, then attack can be failed.

- The minimal samples for a valid hypothesis test is 30:
  - Such test cannot establish on a small fraction samples;
  - However, in *Machine Unlearning*, they start with 1-15 deletion requests.

- The verified unlearning algorithms are limited.