Understanding the Robustness of Machine Unlearning Models

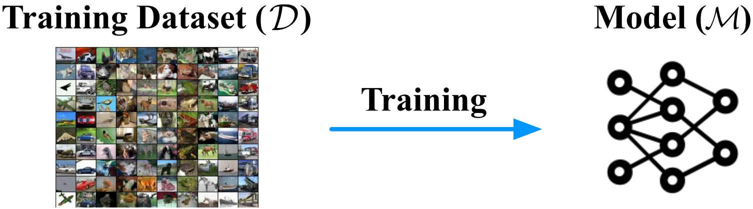
Guanqin Zhang
supervisor: Prof.Yulei Sui

October 21, 2022

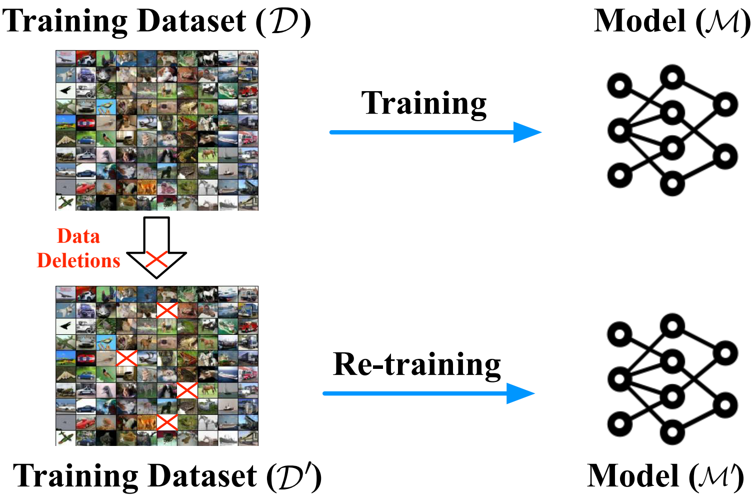
Agenda

- What is machine unlearning?
- Why we need machine unlearning?
- Two unlearning approaches
- Robustness issues
- Findings

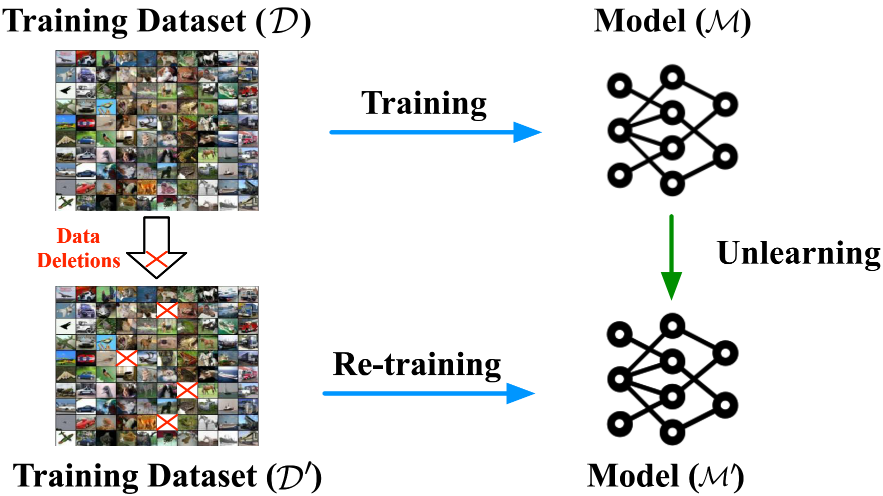
What is Machine Unlearning?



What is Machine Unlearning?



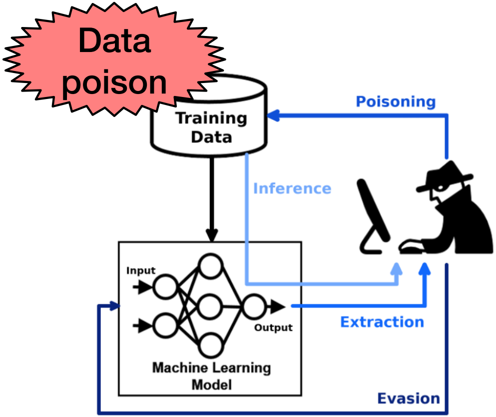
What is Machine Unlearning?



Why We Need Machine Unlearning (1)?

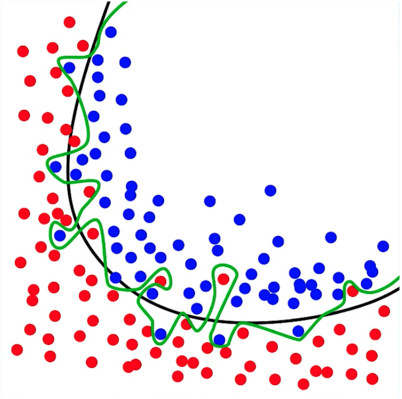
PIPEDA

Why We Need Machine Unlearning (2)?

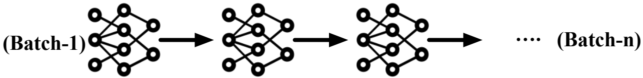


Why We Need Machine Unlearning (3)?

- Overfitting problems
  - Time consuming
  - Overparameterization
  - Overlearnt the features are trivial



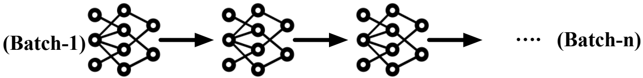
Recent Unlearning Method 1 (Amensiac Unlearning)



- (Training) Model $\mathcal{M}$ is trained for E epochs with B batches. The parameters are updated after each batch by an amount $\Delta_{\theta_{e,b}}$

$$\theta_{\mathcal{M}} = \theta_{initial} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta_{\theta_{e,b}}$$

Recent Unlearning Method 1 (Amensiac Unlearning)



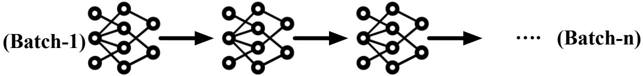
- (Training) Model $\mathcal{M}$ is trained for E epochs with B batches. The parameters are updated after each batch by an amount $\Delta_{\theta_{e,b}}$

  $\theta_{\mathcal{M}} = \theta_{initial} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta_{\theta_{e,b}}$

- (Unlearning) Model $\mathcal{M}'$ uses amensiac unlearning approach that removes the deleted data (belonging to batches) from the learned parameters $\theta_{\mathcal{M}}$

  $\theta_{\mathcal{M}'} = \theta_{initial} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta_{\theta_{e,b}} - \sum_{sb=1}^{SB} \Delta_{\theta_{s,b}}$

Recent Unlearning Method 1 (Amensiac Unlearning)

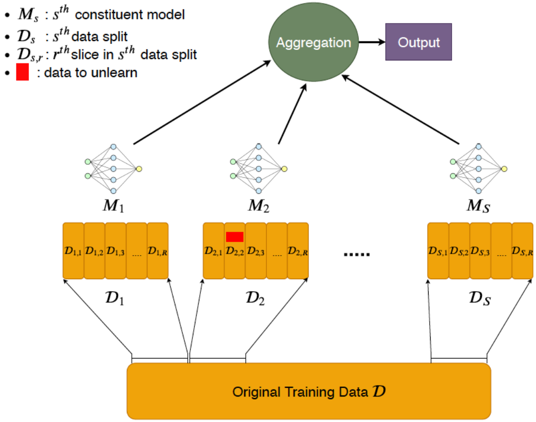


- (Training) Model $\mathcal{M}$ is trained for E epochs with B batches. The parameters are updated after each batch by an amount $\Delta_{\theta_{e,b}}$

$$\theta_{\mathcal{M}} = \theta_{initial} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta_{\theta_{e,b}}$$

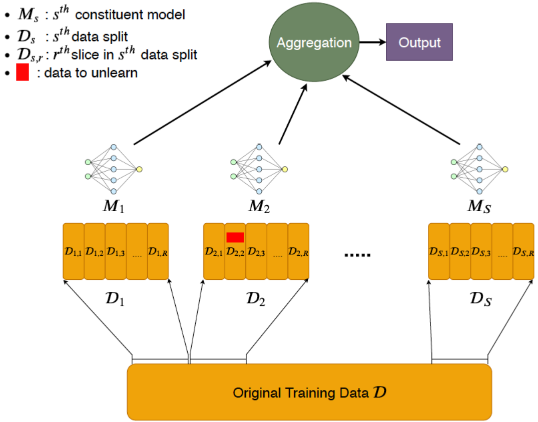
- (Unlearning) Model $\mathcal{M}'$ uses amensiac unlearning approach that removes the deleted data (belonging to batches) from the learned parameters $\theta_{\mathcal{M}}$

$$\theta_{\mathcal{M}'} = \theta_{initial} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta_{\theta_{e,b}} - \sum_{sb=1}^{SB} \Delta_{\theta_{s,b}} = \theta_{\mathcal{M}} - \sum_{sb=1}^{SB} \Delta_{\theta_{s,b}}$$



- (Training) SISA splits the data into S shards multiple slices with K slices as batches for training the machine learning models.
  $\mathcal{M} = \mathcal{M}_1 \diamondsuit \mathcal{M}_2 \diamondsuit \cdots \diamondsuit \mathcal{M}_s$

Recent Unlearning Method 2 (SISA Unlearning)



- (Training) SISA splits the data into S shards multiple slices with K slices as batches for training the machine learning models.
  $\mathcal{M} = \mathcal{M}_1 \Diamond \mathcal{M}_2 \Diamond \cdots \Diamond \mathcal{M}_s$
- (Unlearning) SISA will delete the data in the specific slice and roll the model parameter back to the storage one ($\mathcal{M}_{s',k'-1}$) to retrain the model ($\mathcal{M}_{s'}$) from the slice ($k' - 1$) without data

Summary

Amensiac Unlearning	**SISA-Unlearning**
- Defines unlearning with the respect to **model parameters**	- Defines unlearning with the respect to the level of **algorithms**

Summary

Amensiac Unlearning	**SISA-Unlearning**
- Defines unlearning with the respect to **model parameters**	- Defines unlearning with the respect to the level of **algorithms**
- Directly modify the parameters	- Retrain needed
- Better efficiency (Approximate)	- Exact deletions

Robustness

To guarantee the robustness of the function $\mathcal{M} : \mathbb{X} \to \mathbb{Y}$ is to ensure

$$x \in \mathbb{X} \ \Rightarrow y = \mathcal{M}(x) \in \mathbb{Y}, \tag{1}$$

which involves checking whether input-output relations of the function hold:

$$\mathbb{X} = \{\hat{x} : \|\hat{x} - x\|_{\mathrm{p}} \leq \sigma\}, \tag{2}$$

Perturbed inputs: During the training period, perturbed inputs are commonly imposed or introduced to mislead the learning process. In response to the perturbations, a robust DNN can be formalized as:

$$\forall x \in \mathbb{X}, \hat{x} \in \mathbb{X}, \|x - \hat{x}\|_{\mathrm{p}} < \sigma \Rightarrow \mathcal{M}(\hat{x}) = y \in \mathbb{Y}, \tag{3}$$

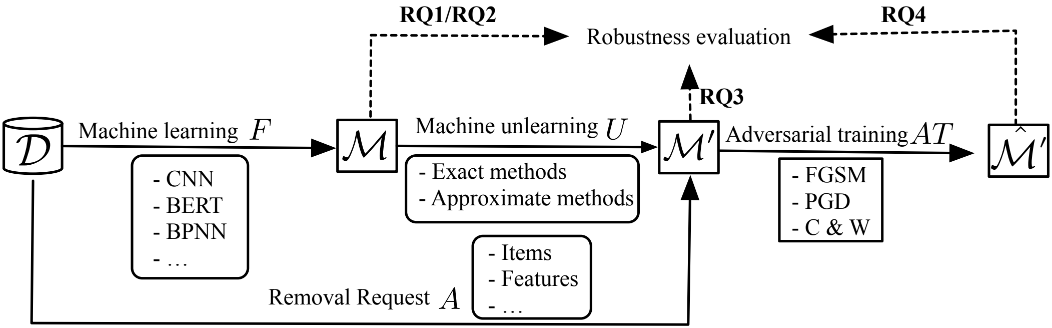
where $\hat{x}$ denotes the perturbed inputs under p normalization with σ distance (the degree of perturbations) to the original input x.

Robustness Issues

$\mathcal{M}'$ is the unlearnt model, we aim to detect whether the retained model is robust

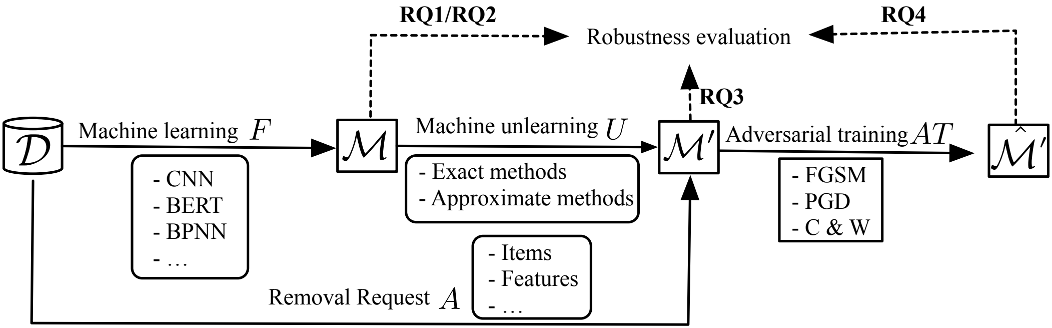
$$\forall x \in \mathbb{X}, \hat{x} \in \mathbb{X}, \|x - \hat{x}\|_{\mathrm{p}} < \sigma \Rightarrow \mathcal{M}'(\hat{x}) = y \in \mathbb{Y}, \tag{4}$$

where $\hat{x}$ denotes the perturbed inputs under p normalization with σ distance (the degree of perturbations) to the original input x.



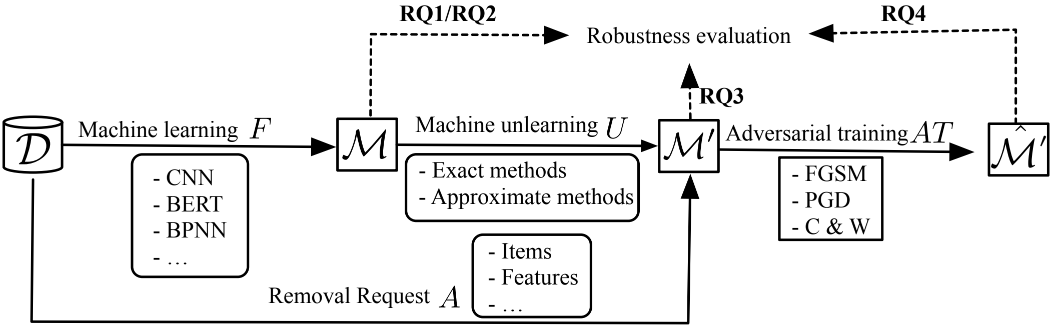
RQ1: How robust is the model by standard training (*st_model*) **before** the data deletion?

Study Framework

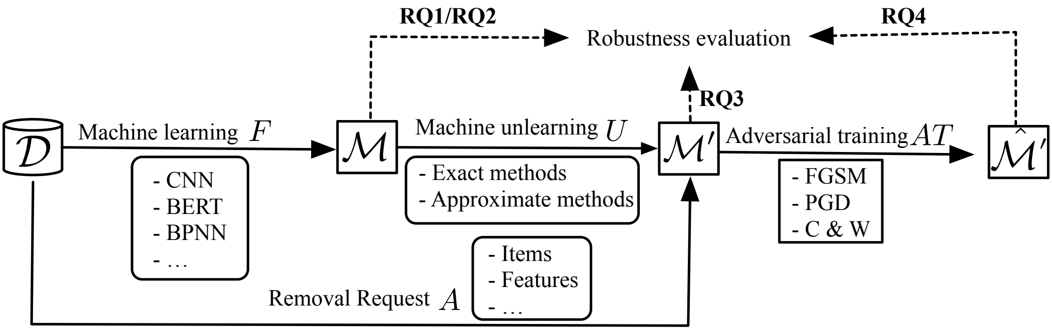


RQ2: How robust is the model by adversarial training (*at_model*) **before** the data deletion?

Study Framework



RQ3:How robust is the (*st_model*) **after** the data deletion ?



RQ4:How robust is the (*at_model*) **after** the data deletion?

Experimental Descriptions

- Loan[1] is a dataset from a US peer-to-peer financial company Lending Club with two subsets, an accepted set containing 2,260,701 instances with 151 columns and a rejected set containing 27,648,741 instances with 9 columns. The dataset is used for loan prediction to get binary results in natural language processing tasks.
- Ham10000[2] is a dataset with 10015 dermatoscopic images of pigmented skin lesions with related information including lesion categories, location of the body, age and gender of patients. The dataset is used for predicting skin lesion types.

[1] N. George, *All lending club loan data*, Apr. 2019. [Online]. Available: https://www.kaggle.com/datasets/wordsforthewise/lending-club.

[2] P. Tschandl, *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*, version V3, 2018. DOI: 10.7910/DVN/DBW86T. [Online]. Available: https://doi.org/10.7910/DVN/DBW86T.

Experimental Descriptions

- Disaster Tweets[3] is a dataset collected from Twitter including 11,000 instances with keywords and location information. The dataset is split into a training set with 7737 instances and a testing set with 3263 instances. The task of the dataset is to predict a binary result that whether a tweet is a disaster tweet.
- Mixture is a dataset synthesizing via a Gaussian mixture model. TBD.

[3] V. S, *Disaster tweets*, Nov. 2020. [Online]. Available: https://www.kaggle.com/datasets/vstepanenko/disaster-tweets.

Experiments Results RQ-1

Dataset	SISA			Amnesiac-ML		
	ACC	R-ACC	Gap	ACC	R-ACC	Gap
Loan	96.53%	31.08%	65.45%	98%	35%	63%
HAM10000	82.23%	21.67%	60.65%	94%	25%	69%
SVHN	96%	43%	53%	95.3%	36%	59.3%
DisasterTweets	94%	53%	41%	95.9%	37%	58.9%
Mini-Imagenet	98%	56%	42%	95.4%	38%	57.4%

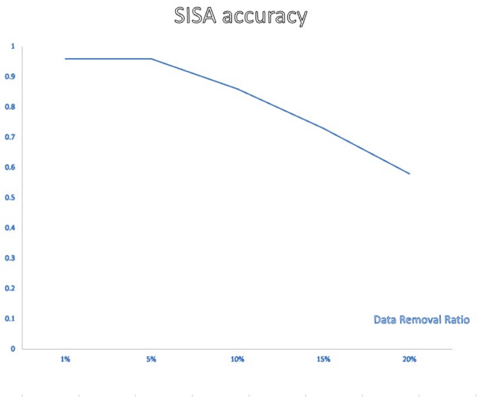
* Perturbations have big impacts on both st_models.

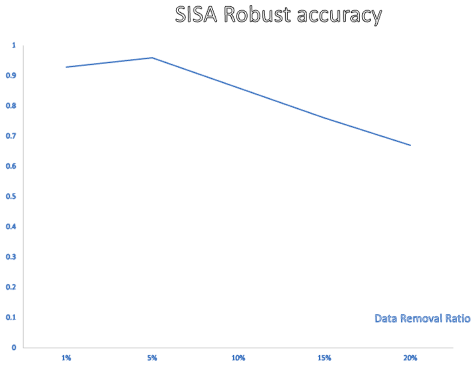
Experiments Results RQ-2

Dataset	SISA			Amnesiac-ML		
	ACC	R-ACC	Gap	ACC	R-ACC	Gap
Loan	94%	88%	6%	90%	86%	4%
HAM10000	80.3%	77.2%	3.1%	90.5%	86.9%	3.6%
SVHN	90.2%	88.6%	1.6%	93.3%	86.4%	6.9%
DisasterTweets	93%	87%	6%	94%	87%	7%
Mini-Imagenet	93%	88.5%	5.5%	93.8%	88.3%	5.5%

* Robust training mitigates the perturbed impacts.

Experiments Results RQ-3 RQ4 - SISA





Experiments Results RQ-3 RQ4 - Amensiac-ML