Trustworthy Machine Learning: the Clash Among the Right to Be Forgotten, Inspectable Models, and Utility

Josep Domingo-Ferrer (*Fellow, IEEE*) ✉ and David Sánchez (*Senior Member, IEEE*)

Universitat Rovira i Virgili,
Department of Computer Engineering and Mathematics,
CYBERCAT-Center for Cybersecurity Research of Catalonia,
ComSCIAM-Center for Computational Science and Applied Mathematics
Av. Països Catalans 26, 43007 Tarragona, Catalonia
{josep.domingo,david.sanchez}@urv.cat

Abstract. Comparing a machine learning model's output before and after unlearning a data point allows an attacker to determine that the data point has been unlearned and hence was a member of the training data. This attack violates the right to be forgotten, and, for inspectable models, it can hardly be prevented without substantial utility loss.

1 Introduction

The right to be forgotten (RTBF), also called the right to erasure, is enshrined in regulations such as the European Union's General Data Protection Regulation [5], the United Kingdom's Data Protection Act [8], the California Consumer Privacy Act [6], and similar laws in Brazil, Argentina, South Korea, Japan, India, and Canada. For several reasons, individuals are entitled to request the deletion of their personal data from various sources (databases, search results, platforms, or data used to train machine learning models, etc.).

In recent years, machine unlearning (MU, [3,2]) has gained a lot of attention as a way to enforce the right to be forgotten in the data used to train a machine learning (ML) model. Paradoxically, MU facilitates membership inference inference attacks (MIAs, [7]), which are privacy attacks that aim to discover whether a target data point was a member of an ML model's training data.

Whereas in a real-world scenario attackers cannot be sure about the success of a MIA because they do not have access to the data actually used to train a model, in the case of MU the success can be checked without access to the training data. Indeed, comparing the model output for a certain data point before and after unlearning gives a clear success criterion for membership. If the class probabilities for that point output by the models before and after unlearning visibly differ, then the point is an unlearned one and hence was a member of the training data for the model before unlearning. We refer to this attack as *differential MIA* (Δ-MIA, see Figure 1).

2 Defenses against the differential MIA

The threat of the Δ-MIA on unlearned data has been largely ignored in the literature. In [4] some defenses have been proposed, which we summarize below:

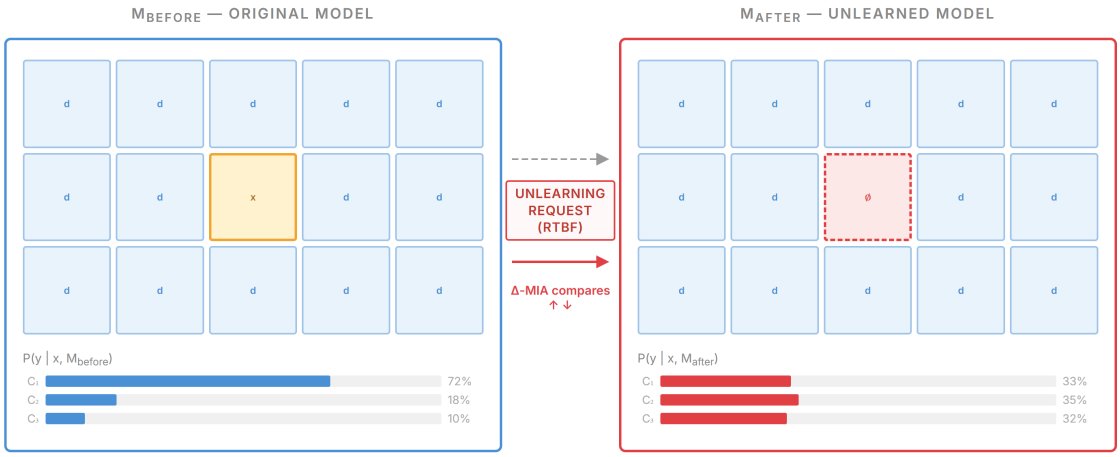


Fig. 1. The machine unlearning paradox. To satisfy the **right to be forgotten** (RTBF) for a data point x, the owner of the model M_{before} produces an unlearned model M_{after}. An attacker who sees both M_{before} and M_{after} can mount a **differential membership inference attack** (Δ-MIA): any shift in output probabilities for x between M_{before} and M_{after} is a deterministic proof that x was in the training data for M_{before}.

1. *Input pre-processing.* After unlearning a certain item, the model keeps a one-way hash image of the unlearned item, and it stores the output class probabilities that were returned for the item prior to unlearning it. At inference time, the model first computes the hash of the input, and, if it matches a hash image of an unlearned item, the model returns the output probabilities stored with the hash image (perhaps with some small perturbation). The goal is to make the behavior of the model for an item indistinguishable before and after unlearning it.
2. *Calibrated perturbation.* Calibrated noise is introduced into the model output during inference after unlearning. The noise is derived from the observed differences in predictions on unlearned items before and after unlearning. To calibrate noise for items not yet unlearned, the unlearning process must be simulated on a random subset of the training data. The goal is to obscure the distinguishability of unlearned items while preserving the utility of the model.
3. *Nearest neighbor confidence calibration.* Direct model outputs during inference after unlearning are mixed with aggregated responses from similar items (the nearest neighbors) in a held-out data set. In this way, the distinguishability between unlearned and retained data is reduced.
4. *Nearest neighbor confidence calibration with KL regularization.* This is a variant of the previous approach. Instead of doing the calibration with neighbors during inference after unlearning, the calibration is performed during unlearning. The unlearning objective of MU is augmented with a Kullback-Leibler (KL) divergence regularization term that guides the model's behavior on unlearned items toward that of similar unseen samples (the nearest neighbors).
5. *Randomized training/fine-tuning.* To thwart a MIA from comparing the behavior of the models before and after unlearning, some randomization in the unlearning process is applied, so that the model after unlearning may behave differently *also for the retained items*. Thus, when the

MIA observes a different behavior for an item before and after unlearning, it cannot tell whether the item was unlearned or retained.

All of the defenses mentioned above except the first one require noise injection and hence result in some utility loss that accumulates after each unlearning round. An additional problem of defenses based on noise calibration (second, third, and fourth) is that the calibration may leak some information on the unlearned item(s) into the model after unlearning. This can erode the unlearning guarantees in case a guaranteed unlearning method was used [1].

The first defense (input pre-processing) can be loss-free if no perturbation is applied to the stored output class probabilities. However, if the attacker observes that exactly the same probabilities are output when a certain item is input twice to the model after unlearning, the attacker can infer that the item has been unlearned and thus that it was a member of the training data set prior to forgetting. Hence, in practice, the first defense also requires a small amount of perturbation and therefore incurs also some utility loss, although the perturbation can be tuned to make this loss really very small.

Beyond potentially offering the smallest utility loss and the smallest information leakage, the pre-processing approach is also the easiest to apply, as it requires no unlearning simulation on data not unlearned (unlike the second defense), it does not require held-out data (unlike the third and fourth defenses), it does not require altering the unlearning process (unlike the fourth and fifth defenses), and it does not affect inferences on the retained data (unlike the fifth defense).

3 Shortcomings of defenses for inspectable models

Inspectable ML models are those whose parameters and weights are fully published. Inspectability is a desirable property in the context of trustworthy AI, as it allows checking for structural transparency, lack of bias, etc., and it is often a requirement in critical applications. There are two types of inspectable models:

- *White-box models.* These are simple models whose parameters are published, such as linear regression, logistic regression, linear classifiers, decision trees, rule lists, naive Bayes, or Bayesian networks. Due to their simplicity, not only can the inspector look at their parameters, but can also interpret them; this means following the model's reasoning path and understanding why a specific output was produced.
- *Inspectable models that are not interpretable.* These are ML models whose parameters are published but are too complex to allow interpretation by humans. In this category, we can find deep decision trees, deep neural networks for prediction, and even some LLMs and other generative models. For example, Meta's LLaMA, LLaMA-2, LLaMA-3, GPT-Neo, and some releases of Mistral are inspectable LLMs.

Inspectability, no matter its type, allows an attacker to bypass the first three defenses mentioned in the previous section. The reason is that those defenses operate at inference time and they are external to the unlearned model. Specifically:

- The input pre-processing can be neutralized without even resorting to the Δ-MIA. Indeed, an attacker could compute the hash of the input and compare it against the stored hash images (which are inspectable): if there is a match, then the input is an unlearned item. Admittedly, this latter attack could be thwarted by modifying the defense to store together with the hash

images and outputs corresponding to unlearned items a much higher number of hash images and outputs corresponding to retained items; in that case, a match would most likely involve a retained item rather than an unlearned one. A more effective attack strategy is to locate the pre-processing code through inspection and skip it when conducting inference. By doing so, the model before unlearning and the model after unlearning will produce different class probabilities for the unlearned inputs. This will allow the differential MIA to succeed.
- The two defenses based on injecting calibrated noise at inference time are also vulnerable to the Δ-MIA. The code injecting noise can be located by the inspector and skipped at inference time. As above, this will cause models before and after unlearning to produce different outputs for unlearned inputs, and thereby facilitate the Δ-MIA.

In contrast, the fourth and fifth defenses above work at the time of unlearning and thus yield unlearned models without external codes that can be located and skipped by the attacker. Thus, models before and after unlearning will produce similar outputs for the unlearned inputs, which thwarts the attack.

Whereas those two defenses do a good job in making inspectability compatible with protection against the Δ-MIA (and hence compatible with the RTBF), they have a utility problem. The reason is that they introduce noise in each unlearning process. This noise accumulates as successive unlearning processes are performed and may render the model unusable after only a few unlearning rounds. For the fifth defense, which distorts the outputs after unlearning for unlearned inputs as well as for retained inputs, this is quite clear: the utility of the unlearned model degrades for all inputs. For the fourth defense, the goal is to distort the output of the model after unlearning only for the unlearned inputs, so that the output should be similar to that of the original model; however, this distortion is likely to propagate to the output of the unlearned model for the retained inputs, because the more complex a model is, the harder it is to confine the distortion to a predefined subset of the inputs.

It may be objected that other defenses against the Δ-MIA might be designed that allow inspectability while providing good utility and protection. Even so, they must rely on either adding code to the unlearned model for it to be run at inference time or distorting the model at unlearning time. No other option seems feasible. As noted above, added code can be skipped by an inspector, and distorted unlearned models result in cumulative utility degradation. The situation is summarized in Figure 2.

4 Conclusions

Trustworthy machine learning seeks to design, train, deploy, and govern ML systems so that humans, institutions, and society can reasonably rely on them.

Satisfying the RTBF by allowing certain personal information to be excluded from an ML model is a property required for a model to inspire trust and is also a legal duty in several jurisdictions, including the European Union.

Other properties that are expected from trustworthy ML are interpretability/explainability, transparency, and fairness. Inspectability is certainly not the only way to achieve those additional properties, but it goes a long way towards satisfying them.

We have highlighted the threat posed by the Δ-MIA, which is commonly neglected in the literature, and we have argued that it is difficult to reconcile ML inspectability with the RTBF and with a good level of model utility. Our aim is to introduce some realism in the expectations placed in trustworthy machine learning: not everything seems to be possible at the same time.

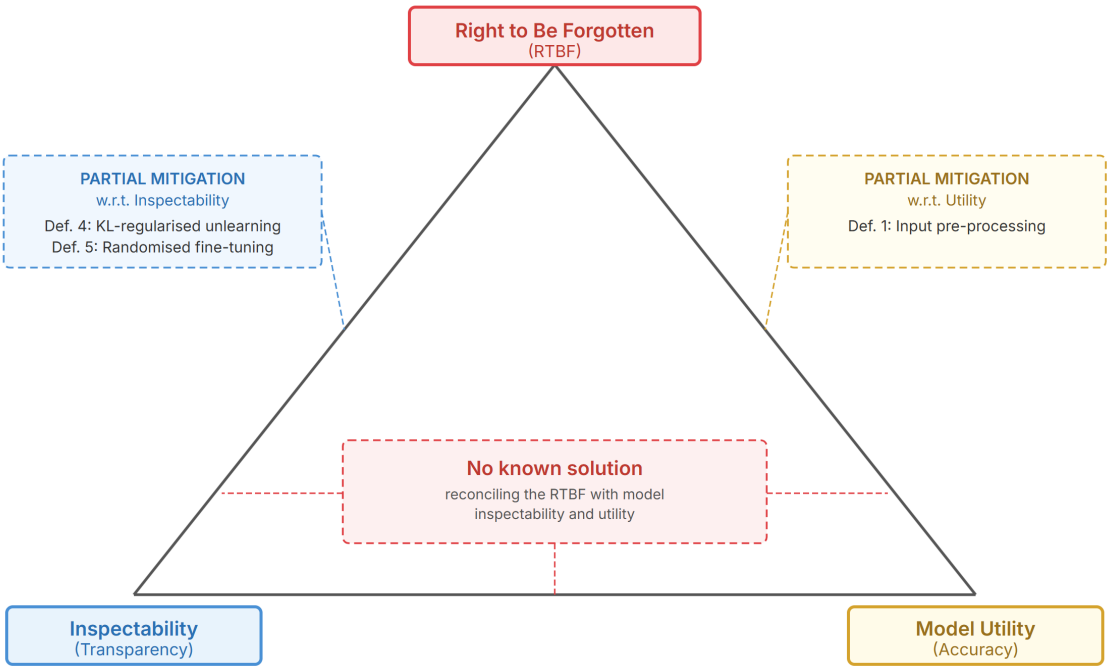


Fig. 2. Three-way **conflict between transparency, the RTBF, and utility**: if models are inspectable, inference-time defenses against the Δ-MIA are bypassable, while unlearning-time defenses inject noise that unacceptably degrades utility over successive unlearning rounds.

Acknowledgments

Partial support to this work has been received from the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and to D. Sánchez) and MCIN/AEI under grant PID2024-157271NB-I00 "CLEARING-IT".

References

1. Alberto Blanco-Justicia, Najeeb Jebreel, Benet Manzanares-Salor, David Sánchez, Josep Domingo-Ferrer, Guillem Collell, and Kuan Eeik Tan. Digital forgetting in large language models: A survey of unlearning methods. *Artificial Intelligence Review*, 58(3):90, 2025.
2. Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.
3. Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.
4. Josep Domingo-Ferrer, Najeeb Jebreel, and David Sánchez. Defenses against membership inference attacks on unlearned data. In *International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2025)*, pages 145–159. Springer, 2025.
5. European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). `https://data.europa.eu/eli/reg/2016/679/oj`, 2016.
6. Parliament of the United Kingdom. Uk data protection act 2018. `https://www.legislation.gov.uk/ukpga/2018/12/contents`, 2018.
7. Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.
8. State of California. California consumer privacy act. `https://oag.ca.gov/privacy/ccpa`, 2018.