The Differential Membership Inference Attack: Defenses and their Shortcomings for Inspectable Models

Josep Domingo-Ferrer and David Sánchez

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. Although the effectiveness of membership inference attacks in real-world conditions has recently been questioned in the literature, there is at least one scenario in which membership can be inferred by an attacker with high certainty: machine unlearning. By calculating the difference between the output of the models released before and after unlearning, the attacker can easily determine whether a data point has been unlearned and, therefore, was a member of the training data before unlearning. This clearly violates the right to be forgotten (RTBF), which paradoxically is the very principle that usually motivates machine unlearning. Some defenses against this issue have been proposed that try to make the output of models before and after unlearning indistinguishable. We argue that most of these defenses can be easily bypassed when the unlearned models are made inspectable —inspectability being regarded as desirable in trustworthy AI. To make matters worse, defenses that withstand such bypassing attacks tend to degrade utility in a sharp way after just a few rounds of unlearning. This signals an unsolved conflict among the RTBF, inspectability, and model utility.

Keywords: Machine unlearning · right to be forgotten · Machine learning inspection · membership inference · privacy.

1 Introduction

Membership inference attacks (MIAs, [18]) aim to determine whether a given data point was included in the data set used to train a certain machine learning (ML) model. MIAs are also instrumental as building blocks of more sophisticated privacy attacks, such as attribute inference or reconstruction.

Although recent literature [13,12] has questioned the effectiveness of MIAs in real-world conditions, there is at least one situation in which there is a clear success criterion for membership inference: machine unlearning (MU, [3,2]). In recent years, MU has gained a lot of attention as a way to remove unwanted

or problematic information from a trained model in a more efficient way than retraining the model from scratch on the data to be retained. Forgetting some types of data, such as personal or copyrighted data, is also a legal requirement. In such scenarios, unlearning personal information helps comply with the right to be forgotten (RTBF) —enshrined in regulations such as the European Union's General Data Protection Regulation [9], the United Kingdom's Data Protection Act [19], the California Consumer Privacy Act [15], and similar laws in Brazil, Argentina, South Korea, Japan, India, and Canada—, while unlearning copyrighted information helps comply with copyright laws.

Paradoxically, MU facilitates membership inference for unlearned 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 item before and after unlearning gives a clear success criterion for membership. If the class probabilities for that item output by the models before and after unlearning visibly differ, then the item 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).

In this paper, we formalize the Δ-MIA attack in the MU scenario, review possible defenses against it, and discuss their shortcomings with respect to model utility and also inspectability (a desirable property in trustworthy AI). Our analysis signals an unsolved conflict among the RTBF, model inspectability and model utility.

The remainder of this paper is organized as follows. Section 2 surveys the MU methods, characterizing them according to the type of guarantees of forgetting they offer. Section 3 formalizes the Δ-MIA. Section 4 reviews the defenses against this attack and Section 5 discusses the limitations of the defenses reviewed. The final section gathers the conclusions of our analysis.

2 Machine unlearning methods

Depending on the type of forgetting guarantees they offer, MU methods can be classified as exact, privacy-guaranteed through pre-protection, certified, and approximate (see Figure 1).

Exact unlearning [3,2,23] aims to completely eliminate the influence of data items to be forgotten by using (re-)training strategies that are more efficient than retraining from scratch. A prominent approach in this class is SISA [2], which divides the training data into S shards of R slices and trains a partial model for each shard slice-by-slice with checkpoints. Upon completion, partial models are saved within their respective shards. During inference, the outputs are aggregated in a ensemble style. For data forgetting, the slice containing the data to be forgotten is identified, the target data are removed, and the model of the corresponding shard is retrained from the last checkpoint to ensure effective unlearning. The shortcomings of exact forgetting are that the whole procedure is quite costly —especially if items from the first slices of a shard need to be

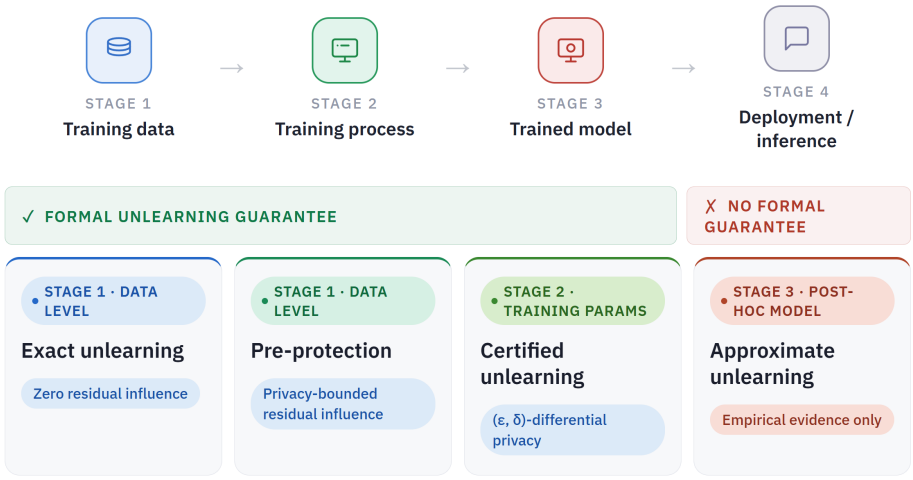


Fig. 1. Types of machine unlearning methods and their guarantees across the ML model lifecycle

forgotten—, and that ensembling partial models may omit semantic nuances that are critical in large language models [1].

Unlearning with privacy guarantees through pre-protection [7] tries to alleviate the cost of exact unlearning while still offering a guarantee at a data level by enforcing privacy protection on the training data *before* model training. First, a version of the training data is protected using a privacy guarantee. Any privacy guarantee can be used, as long as it is immune to postprocessing; such is the case of probabilistic k-anonymity and ϵ-differential privacy [8]. Second, an ML model is trained on the protected training data to obtain a privacy-protected —but probably inaccurate— pre-trained model. Finally, the pre-trained model is fine-tuned on the original training data —thereby dropping the privacy guarantees— to recover model accuracy, and it is released for production. To unlearn some data, the protected model before fine-tuning is recovered, and it is fine-tuned on the original data minus the data to be forgotten. In this way, the data to be forgotten remain protected by the privacy guarantee in use. This approach is more efficient than exact unlearning because fine-tuning is cheaper than re-training from scratch on the data to be retained. In addition, the utility of the model after unlearning is better preserved because the forgotten data are privacy-protected rather than completely removed.

Certified unlearning also employs a privacy guarantee ((ϵ, δ)-differential privacy), but rather than enforcing it at the training data level, it acts at the model parameter level to probabilistically limit the influence of the data to be forgotten on the model output. The certified unlearning schemes in the literature [10,17,5,25,4,16,14] combine an approximate unlearning step (such as a gradient update) with calibrated noise addition to the weights to satisfy the target pri-

vacy guarantee. Since most of them assume convexity or strong convexity, their applicability to neural networks is limited. Moreover, since they operate at the model parameter level, their guarantees are generally less precise and harder to interpret than those of the two data-level approaches discussed above.

Approximate unlearning aims to provide a more efficient alternative by minimizing the influence of unlearned items to an acceptable level through an *ex post* alteration of the learned model [22]. Approximate unlearning methods do not offer formal guarantees of data removal (thus falling short of legal compliance), and evidence of removal is at best empirical [11,20,24,21].

3 The differential MIA

The guarantees of unlearning achieved by all types of unlearning (except approximate unlearning) focus on ensuring that after unlearning, M_{after} no longer remembers the data to be forgotten. That would be enough if only the model after unlearning (M_{after}) was available to users and potential attackers. However, although keeping a copy of past models is costly in terms of storage, it is quite feasible for potential attackers to store all the versions of a model released by the model owner after successive unlearning operations. Therefore, mounting a Δ-MIA attack is quite possible, and any guarantees of exact, privacy-guaranteed via pre-protection, or certified unlearning do not prevent this attack.

Formally, given an initial ML model M_{before} trained on a data set $\mathbf{D}$ and one or several data points $\mathbf{D}_f$ to be forgotten, MU transforms M_{before} into a new model M_{after} that is (nearly) as ignorant of $\mathbf{D}_f$ as a model trained on $\mathbf{D} \setminus \mathbf{D}_f$. The Δ-MIA works as follows (see Figure 2). An attacker can compare the class probabilities output by M_{before} and M_{after} for a certain data point x. If the class probabilities output by M_{before} for x, say $M(x)$, differ from the class probabilities output by M_{after} for x, say $M'(x)$, that is, if $|M(x) - M'(x)| > T$ for appropriate distance $|\cdot|$ and threshold T, then the attacker can conclude with reasonable confidence that $x \in \mathbf{D}_f$ and therefore that x was really a member of the training data of M_{before}.

4 Defenses against the differential MIA

The threat of the Δ-MIA on unlearned data has been largely ignored in the literature. In [6] some defenses have been proposed, which we summarize below:

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 it is unlearned.

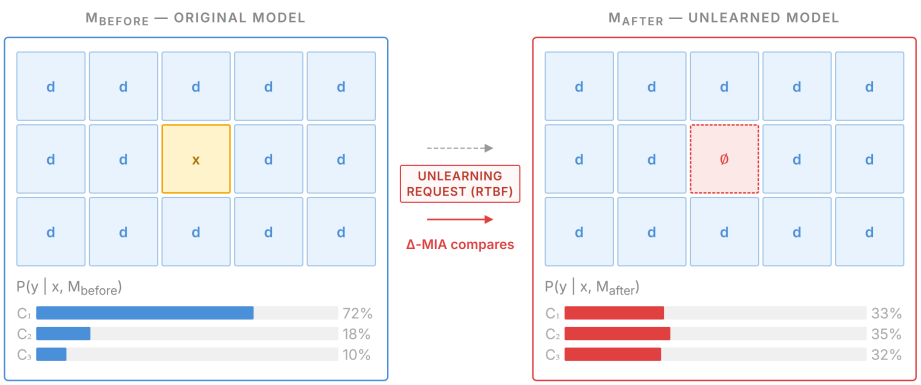


Fig. 2. An attacker who sees both M_{before} and M_{after} can mount a *differential membership inference attack* (Δ-MIA): any significant shift in output probabilities for x between M_{before} and M_{after} is a deterministic proof that x was in the training data for M_{before}.

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 prevent an MIA from comparing the behavior of the models before and after unlearning, some randomization is applied in the unlearning process, so that the model after unlearning may also behave differently *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 (*e.g.,* exact, privacy-guaranteed via pre-protection, or certified).

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).

5 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:

- *Interpretable models.* Also known as 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 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 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 the injection of 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 M_{after} without external codes that can be located and skipped by the attacker. Thus, M_{before} and M_{after} 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 of M_{after} for unlearned inputs and also for retained inputs, this is quite clear: the utility of M_{after} degrades for all inputs. For the fourth defense, the goal is to distort the output of M_{after} only for the unlearned inputs, so that the output should be similar to that of M_{before}; however, this distortion is likely to propagate to the output of M_{after} 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. However, they must rely on either adding code to M_{after} to be run at inference time or distorting M_{after} 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.

6 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

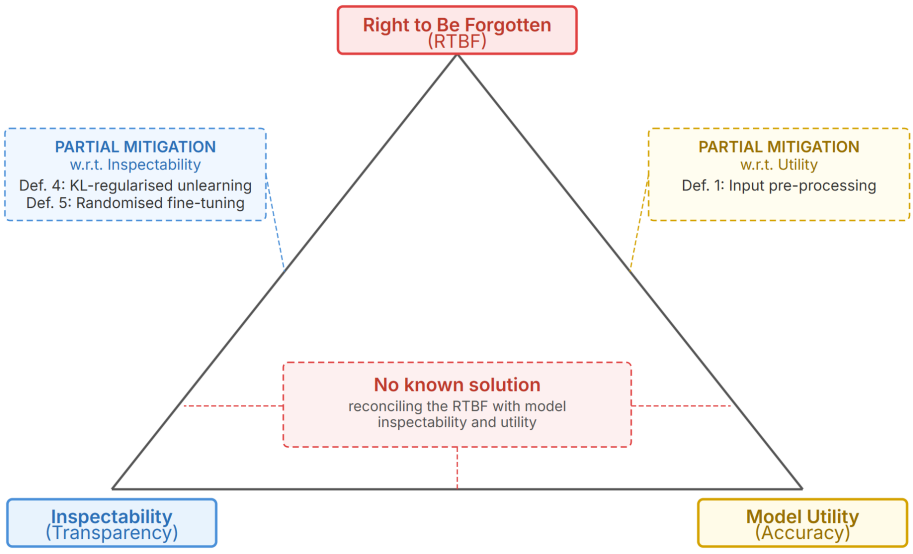


Fig. 3. Three-way conflict between transparency, the RTBF, and utility: if models are inspectable, inference-time defenses against the Δ-MIA are bypassable, while unlearning-time defenses inject noise that unacceptably degrades utility over successive unlearning rounds.

a legal duty in several jurisdictions, including the European Union. Similarly, being able to exclude copyrighted content from the training data is necessary to comply with the copyright laws.

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.

In this paper, 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 (see Figure 3). Our contribution is meant to introduce some realism in the expectations placed in trustworthy machine learning: not everything seems to be possible at the same time.

Acknowledgments

Partial support to this work has been received from the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and David Sánchez), Agencia Española de Investigación (project PID2024-157271NB-I0 “CLEARING-IT”), and the European Commission (project HORIZON-101292277 “SoBigDataIP”).

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. Eli Chien, Haoyu Peter Wang, Ziang Chen, and Pan Li. Certified machine unlearning via noisy stochastic gradient descent. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
5. Rishav Chourasia and Neil Shah. Forget unlearning: Towards true data-deletion in machine learning. In *International conference on machine learning*, pages 6028–6073. PMLR, 2023.
6. 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.
7. Josep Domingo-Ferrer, Najeeb Jebreel, and David Sánchez. Efficient unlearning with privacy guarantees. *CoRR*, abs/2507.04771, 2025. https://doi.org/10.48550/arXiv.2507.04771.
8. Josep Domingo-Ferrer and David Sánchez. How to get actual privacy and utility from privacy models: the k-anonymity and differential privacy families. *Transactions on Data Privacy*, To Appear, 2026.
9. 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.
10. Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020*, pages 3832–3842, 2020.
11. Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2021.
12. Najeeb Jebreel, Mona Khalil, David Sánchez, and Josep Domingo-Ferrer. Revisiting the lira membership inference attack under realistic assumptions. In *26th Privacy Enhancing Technologies Symposium - PETS 2026*, To Appear.
13. Najeeb Jebreel, David Sánchez, and Josep Domingo-Ferrer. A critical review on the effectiveness and privacy threats on membership inference attacks. In *31st European Symposium on Research in Computer Security - ESORICS 2026*, To Appear.
14. Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In *Forty-second International*

Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025, 2025.
15. Parliament of the United Kingdom. Uk data protection act 2018. `https://www.legislation.gov.uk/ukpga/2018/12/contents`, 2018.
16. Xinbao Qiao, Meng Zhang, Ming Tang, and Ermin Wei. Hessian-free online certified unlearning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.
17. Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.
18. 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.
19. State of California. California consumer privacy act. `https://oag.ca.gov/privacy/ccpa`, 2018.
20. Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
21. Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. In *Proceedings 2023 Network and Distributed System Security Symposium*. Internet Society, 2023.
22. Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.
23. Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, volume 6, page 19, 2022.
24. Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. How does data augmentation affect privacy in machine learning? *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10746–10753, 2021.
25. Binchi Zhang, Yushun Dong, Tianhao Wang, and Jundong Li. Towards certified unlearning for deep neural networks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.