Comparative Analysis of Machine Unlearning Approaches for Data Protection

Maria Salgado Herrera
ThereSIS
Thales SIX GTS France
Palaiseau, France
msalgadoh@unal.edu.co

Vincent Thouvenot
ThereSIS
Thales SIX GTS France
Palaiseau, France
vincent.thouvenot@thalesgroup.com

Alice Héliou
ThereSIS
Thales SIX GTS France
Palaiseau, France
alice.heliou@thalesgroup.com

Adrien Becue
ThereSIS
Thales SIX GTS France
Gennevilliers, France
adrien.becue@thalesgroup.com

***Abstract*—The increasing sensitivity and widespread adoption of machine learning models in various applications have led to a growing need for "machine unlearning" - the process of removing the influence of specific data used to train a model without retraining from scratch the model. As machine learning models handle personal and sensitive data, it is crucial to develop robust and adaptive unlearning approaches that can defend against attacks. This paper provides a comprehensive overview of the latest tools and approaches used in machine unlearning, focusing on short execution time and good performance. We present a standardized comparison of different automatic unlearning methods, highlighting their differences, advantages, and limitations. Our work aims to contribute to the development of more efficient and effective machine unlearning techniques, addressing the challenges of user privacy, bias removal, and confusion resolution.**

***Index Terms*—Deep neural networks, Image classification, Machine Learning, Security, Privacy, Machine Unlearning**

I. Introduction

Machine learning models emerged with the objective of training a dataset to learn parameters and create relationships in the data. This process involves feeding the model with a substantial amount of data, allowing it to recognize patterns and correlations. Once the algorithm is created, it can be used to make predictions on some input data. The goal of machine learning (ML) is to create accurate models that can generalize well to new data [1].

In recent years, machine learning models have evolved significantly, becoming increasingly sensitive and specific in their capabilities. These models are now commonly used in a wide variety of applications that involve critical information about users, tasks, and processes. For example, in sectors such as biometrics [2] and security in Cyber Physical systems, such as electric power grid [3], machine learning models handle personal and/or sensitive data that require careful and ethical management. With the increase in the sensitivity of these models and the amount of personal data they process, a new need has arisen: machine unlearning. Machine unlearning refers to the process of selectively removing specific data points from a machine learning model, to effectively ”forget” information.

Moreover, the development of automatic unlearning models is constantly growing and a common challenge is their defense against attacks, so many approaches have been proposed to develop models that are as adaptive and robust as possible. This has fueled a debate on the feasibility of approximate unlearning, which would enable models to forget learned patterns without requiring complete retraining from scratch, thereby significantly reducing the computational costs. The adoption of approximate machine unlearning techniques over exact machine unlearning is justified by their superior efficiency and scalability, as well as their adequacy for numerous applications. Although exact machine unlearning, which entails manual removal of the information to be forgotten and retraining from scratch, achievies ideal results, its implementation is often hindered by high computational costs and time requirements. In contrast, approximate machine unlearning offers a more practical and effective solution for mitigating the influence of unwanted data in a model, particularly in scenarios where complete removal of the unwanted data's influence is not feasible or necessary [4].

In this paper, we will focus on explaining the development of different automatic approximate unlearning approaches with the aim of revealing the differences and possible advantages depending on the tools used to perform the forgetting of the information used by the model to be forgotten. Our contribution consists in showing in a standardized way an overview of the latest tools used in information forgetting in machine learning models, with a focus on short execution time while maintaining a good performance. Additionally, we seek to evaluate the robustness of the models through attacks that typically aim to exploit vulnerabilities, with the goal of maintaining a good performance. Our paper is structured in 7 sections. Following this introduction, Section 2 provides insights on business needs and motivations, Section 3 introduce the theoretical background, Section 4 describes the methodology, Section 5 establishes the experimental datasettings, Section 6 provides study results and Section 7 concludes.

II. Business need/Motivations

Machine unlearning refers to the process of selectively removing specific data points from a machine learning model, effectively ”forgetting” information. This ability is crucial in the defense and security sector, where sensitive data may need to be purged from models for various reasons such as:

the revocation of classified information [1], compliance with data protection regulation [5], mitigation of data poisoning attacks [6], controlled information disclosure [7], updating operational intelligence [8], demilitarization of dual use technology [9] or compliance with export control regulation [10]. While the concept is promising, there are several gaps and challenges in the current state of the art: limited efficiency and scallability [11], insufficient accuracy and integrity [11], confidentiality and security concerns [1], legal and compliance issues [12], limited versatility and generalization [13], lack of theoretical foundations [1]. Security concerns regarding unlearned models, particularly the potential retrieval of "forgotten" data classes, revolve around the risk that sensitive or protected information may still be inferred or reconstructed, even after attempting to remove it [11]. For these reasons, a comparative assessment of diverse unlearning techniques aiming to evaluate their robustness to membership inference attacks is of paramount importance for applications in the defense and security sector. Moreover, to justify the use of unlearning techniques, rather than retraining techniques, the run-time efficiency of these different approaches needs to be assessed.

III. Background

Machine Unlearning, introduced in 2015 by [14], is an emerging topic of Machine Learning. The unlearning approaches can be approximate or exact. For the latter the resources and time required can be dissuasive. In [7] and [15] , the authors propose a framework for unlearning that relies on data shading and slicing to reduce the computational overhead of unlearning. Such approaches ask to create several sub samples to limit the impact of forgetting an observation. However, it can still be costly if the amount of information to forget is high.

Most of the researches in the field tend to design a quick and efficient way to perform unlearning that approximate as well as possible a retraining from scratch. [16] propose the SCRUB method, that see the unlearning problem as a teacher-student problem. Moreover, they illustrate several tasks of unlearning, in particular removing bias, resolving confusion and user privacy. [17] propose a method called Amnesias Unlearning, where during a training, the model owner keeps a list of which examples appeared in which batches as well as the parameter updates from each batch. To remove a sensitive information, the model owner undoes the parameter updates coming from the batch. [18] propose a linear filtration for logit-based classifier.

Recent research on machine unlearning, such as [19] has identified several key approaches for unlearning requests, including data removal, feature removal, class removal, task removal, and stream removal. Among the advanced techniques, the data remover and class remover are particularly emphasized. The data remover technique unlearns specific data points from the training dataset, crucial for privacy compliance and error correction. The class remover technique eliminates entire classes of data, useful for maintaining relevance and ethical integrity when certain categories are no longer needed or pose ethical concerns [1]. Our paper focuses on data removal.

Verification of unlearning performance is a complex topic (see [20]). As machine learning models have advanced in complexity, the attacks targeting these models have also evolved, aiming to undermine both their security and privacy. Two of the most notable types of attacks are Model Inversion Attacks and Membership Inference Attacks [1]. Model Inversion Attacks focus on reconstructing sensitive features of the training data by exploiting correlations with the model's output, ultimately aiming to recreate the input data, particularly the most private features [21]. Membership Inference Attacks [22] are a method used by adversaries to determine whether a specific example was part of a model's training dataset. These attacks are particularly useful in verifying if a group of data has been successfully forgotten by a model. If unlearning is effective, the attacker should not be able to guess whether a data was forgotten or never used (part of the test dataset). The attack (e.g. [23]) involves identifying whether a particular datum was included in the training dataset, often assessed using "average case" scenarios, where the likelihood that a data point belongs to the target dataset is compared against the test dataset. Key metrics such as true positive rate and false positive rate are crucial in evaluating the effectiveness of these attacks [24].

IV. Methodology

A. Unlearning task and type of model analyzed

1) Unlearning task: Based on the challenge proposed by [20], we consider the challenge illustrated by Figure 1. First, a deep neural classifier, called original model, is trained on a training dataset. From this original model, we train a new model, called unlearned model, whose objective consist in forgetting part of the information while preserving the performance on the other part of the data. During this second stage, the training dataset is divided in two datasets:

- The forget dataset, for which the aim is for the unlearned model to output predictions that are indistinguishable from the predictions made for the test dataset.
- The retain dataset, for which the aim is for the unlearned model to obtained on it similar performance as the original model.

In Figure 1, the test dataset correspond to data never seen during the training of the original and the unlearned models.

Then, information are extracted from the outputs of the unlearned model to evaluate if the outputs for data from forget dataset is closed to the ones of the test dataset.

The evaluation process is explained in Section V-C.

2) Model analyzed: a ResNet18: We consider the model ResNet18 [25]. It is a deep convolutional neural network used for image classification. It contains 18 main layers, including convolutional 2D layers (called Conv2D below), pooling layers and fully connected layers. 2D convolutions start with a kernel, which is simply a small matrix of weights. This kernel "slides" over the 2D input data, performing an elementwise multiplication with the part of the input it is currently on, and

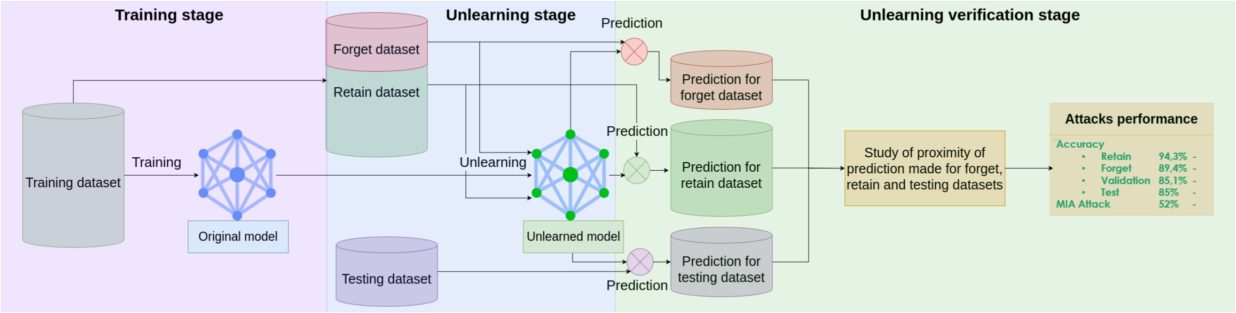

FIG. 1: Unlearning challenge addressed in the paper.

then summing up the results into a single output pixel. Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Moreover, in ResNet18, there are several residual blocks, each consisting of few convolutional layers. These blocks help mitigate vanishing gradient problem [26].

B. Unlearning approaches studied

The models presented in this section were approaches conceived by participants in the NeurIPS 2023 challenge [20], which considers a realistic scenario in which an age predictor has been trained on private face images, and, after training, a certain subset of the training images must be forgotten to protect the privacy or rights of the individuals concerned [20]. The datasets used in the kaggle competition have not been disclosed, so we have studied, adapted and compared theses approaches on another dataset, CIFAR10 [27].

1) Distillation approach [28]: The distillation model is based on the machine learning technique known as knowledge distillation, which is used to transfer knowledge from a large and complex deep neural network (referred to as the “teacher model”) to a smaller and more efficient model (referred to as the “student model”) [29], as illustrated in Figure 2.

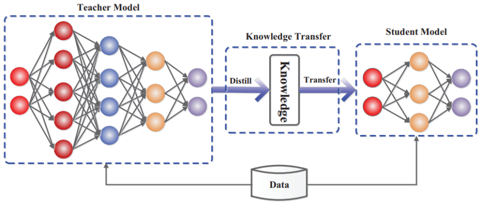

FIG. 2: The generic teacher-student framework for knowledge distillation [29].

In this approach (see Figure 3) after a partial re-initialisation of the model weights, the main idea is to directly mimic the final prediction of the teacher model on the retain dataset. The aim is to maintain the high accuracy and performance of the original model for the information we wish to retain, while reducing the model’s capacity to correctly predict the group of data that has been decided to be forgotten.

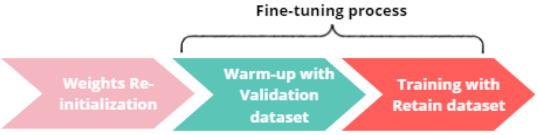
Fine-tuning process
Weights Re-initialization
Warm-up with Validation dataset
Training with Retain dataset

FIG. 3: Steps for the distillation approach.

The initial phase involves resetting the first and last layers of the original model. These two layers are chosen because the first layer significantly influences the rest of the model layers, and the last layer determines the model’s final output distribution. With this reset step, we enable the model to deviate from its original state.

The next step will be fine-tuning the model, starting with a warm-up process where the knowledge distillation can be evidenced by obtaining the predictions of the teacher model with respect to the data of the validation dataset and proceed to use the Kullback-Leibler (KL) divergence loss function, or relative entropy, normally used to compare two data distributions corresponding to predicted data and true labels [30]. In this case the teacher information is considered as true labels and is compared with the student’s prediction, allowing the student to get as close as possible to it. Continuing with the fine tuning on the retain dataset, the same knowledge distillation procedure performed previously is carried out, but in this case soft predictions will be used and not hard predictions. The predictions will be used to feed the loss in three different ways: Soft Cross entropy Loss, Cross Entropy loss [31] and KL-loss. This final procedure allows to maintain the performance of the original model.

2) Rotate approach [32]: The rotation approach involves retraining the model using a modified version of the original model, maintaining high accuracy and performance. The process is divided into two important steps, as shown in the Figure 4.

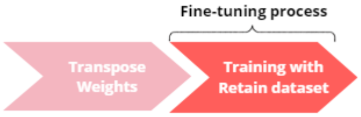

FIG. 4: Steps for the rotate approach.

Initially, the model will undergo unlearning with a modification that involves transposing all weights in Conv2D layers. This process helps in forgetting samples in the forget-set, enabling the reuse of valuable features from the original model. Finally, to refine the model, fine-tuning is performed using Cross Entropy Loss [31].

3) Pseudolabeling approach [32]: The pseudolabeling approach involves retraining the model using a modified version of the retain dataset taking into account the performance of the forget dataset with the original model. The process carried out by this approach, which can be seen in the Figure 5, starts with first, they store the inference result on the forget dataset using the original model in three different ways:

- Store the inference of the original model on the forget dataset;
- Perform a naive unlearning by fine tuning the original model on the retain dataset alone and store the inference of this model on the forget dataset;
- Re-initialize the original model and retrain it on the retain dataset during a few epochs and store the inference of the retrained model on the forget dataset.

Afterwards they define pseudo labels for the forget dataset, such that data on which the classification is not quickly learned from scratch are defined erroneously. Pseudo-labels are set to the predictions of the fine tuned model except when the fine tuned model is correct and the retrained from scratch is wrong with a low logits entropy. In that case, pseudo labels are set to the retrained from scratch model predictions.

Finally, they re-initialize the weight of the Linear layer of the original model and train it on the forget dataset pseudo-labeled using Cross Entropy Loss [31].

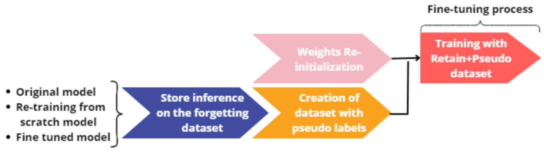
Fine-tuning process
Original model
Re-training from scratch model
Fine tuned model
Store inference on the forgetting dataset
Weights Re-initialization
Creation of dataset with pseudo labels
Training with Retain+Pseudo dataset

FIG. 5: Steps for the pseudolabeling approach.

This approach was combined with the Rotate approach, describe above in IV-B2.

4) Pruning approach [33]: In this approach, the unlearning is achieved by increasing the sparsity of the model, guided by the data pruning process, which consists of identifying and removing unnecessary weights and connections from the model [34]. This approach is performed in two main steps, the first one of weight re-initialization using weights pruning and the second one fine tuning the model, as shown in the Figure 6.

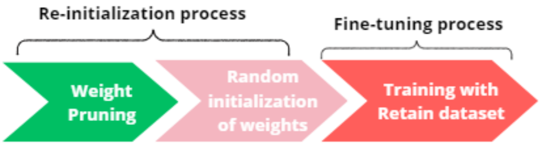
Re-initialization process
Fine-tuning process
Weight Pruning
Random initialization of weights
Training with Retain dataset

FIG. 6: Steps for the pruning approach.

Regarding the first step of re-initialization we will increase the dispersion on Conv2D and Linear layers. For this, the weights corresponding to these layers are collected, then the pruning criterion L1 Unstructured is chosen, in which the L1 norm of each weight is calculated. They are listed in ascending order and the smallest values are eliminated. In the approach 99% of the weights were chosen to be set to zero, and only 1% of the most relevant weights for the model are kept. A random reset of these zero weights is then performed. Finally, following the same process as previous approaches, we proceed with fine-tuning on the retain dataset, combining Cross Entropy Loss [31] and Custom Loss (Mean squared Error (MSE) on logits entropy) [31].

5) Deviation approach [35]: This approach consists of preserving, on average, the global information present in the original trained model, while introducing noise to the model weights.

This approach is performed in two main steps, first by deviating randomly the parameters of the convolutional layers from their real state, then performing fine tuning on the retain dataset.

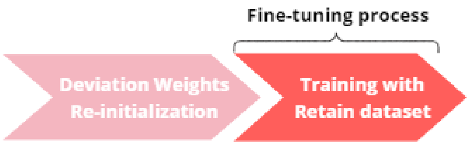
Fine-tuning process
Deviation Weights Re-initialization
Training with Retain dataset

FIG. 7: Steps for the deviation approach.

In the first step, we replace each weight w_i of the convolutional layers by an observation taken from the Gaussian distribution with expectation w_i and standard deviation σ. σ, is chosen arbitrarily at 0.6. This approach aims to mimic the inherent uncertainty in the weights of the neural network from the outset, providing a statistical basis that can influence the convergence and overall performance of the model during training. For the second step, we continue the training process on the retain dataset using an optimizer Stochastic gradient descent (SGD) algorithm [36] and the Cross Entropy Loss. Besides, before starting the final training epoch we introduce a little noise in the weights, in this case using $\sigma = 0.005$. This additional perturbation helps prevent the optimizer from getting stuck in local minima of the loss function by modifying the values of the model's weights.

6) Gradient approach [37]: This approach aims to use the forget dataset to analyze how it influences the gradients in comparison with the retain dataset similarly to the Single-shot Network Pruning (SNIP) method [38]. The gradients collected are used as input to perform the re-initialization of the model. This approach is performed in three main steps.

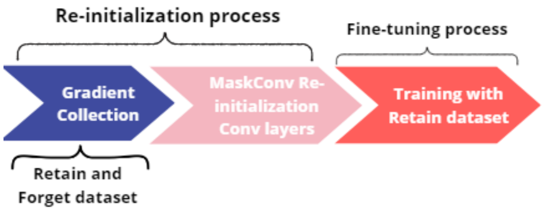
Re-initialization process
Fine-tuning process
Gradient Collection
MaskConv Re-initialization Conv layers
Training with Retain dataset
Retain and Forget dataset

FIG. 8: Steps for the gradient approach.

The first step consists in analyzing the gradients obtained from the predictions of both the forget dataset and the retain dataset. They uses the cross-entropy loss and compare the gradient descent on the retain dataset with the gradient ascent on the forget dataset (see illustration on Figure 9).

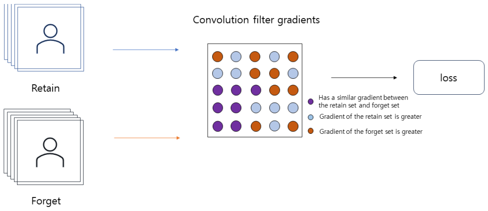
Retain
Forget
Convolution filter gradients
loss
Has a similar gradient between the retain set and forget set
Gradient of the retain set is greater
Gradient of the forget set is greater

FIG. 9: First Step for the gradient approach [37].

Based on the gradient information collected in the first step, for each convolution filter a mask is created to keep only the 30% of its most similar gradients (see illustration on Figure 10). The weights corresponding to the most similar gradients are re-initialized using HE Initialization [39]. Besides, the convolutions are replaced by MaskConv [40] using these masks, in order allow to focus the training on the selected weights.

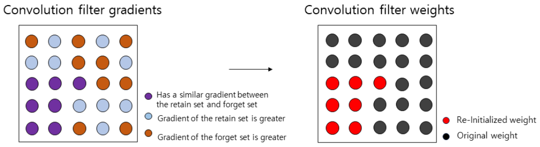
Convolution filter gradients
Has a similar gradient between the retain set and forget set
Gradient of the retain set is greater
Gradient of the forget set is greater
Convolution filter weights
Re-Initialized weight
Original weight

FIG. 10: Second Step for the gradient approach [37].

Finally, following the same process as previous approaches, we proceed with fine-tuning on the retain dataset, using Cross Entropy Loss with Linear Scheduler. The cosine annealing scheduler was also used but did not provide better results for the kaggle competition nor in our experiment

7) Divergence approach [41]: This approach aims to use the forget dataset, to do both the forgetting of it and to do the fine tuning. It is composed of three fundamental steps described in Figure 11.

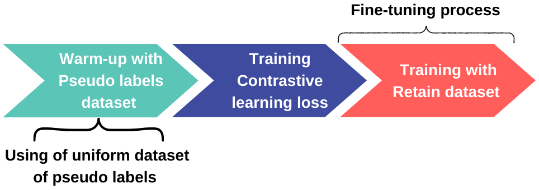
Fine-tuning process
Warm-up with Pseudo labels dataset
Training Contrastive learning loss
Training with Retain dataset
Using of uniform dataset of pseudo labels

FIG. 11: Steps for the divergence approach.

In this case, the first step is a warm-up process that involves training the model on the forget dataset to maximize the divergence between the model's predictions and the correct labels. This will be achieved training the model using KL-loss between output logits and uniform pseudo labels, which are defined as ones.

Finally in the process to maximize prediction accuracy, each epoch involves two key procedures. First, we apply the forget and retain datasets to implement Contrastive Learning Loss [42]. This method refines the model by adjusting the relational distance between the samples in these datasets, either by pushing them apart or pulling them closer together. The second procedure is a fine tuning phase o the retain dataset using Cross Entropy Loss. This dual-step ensures optimal adjustments for improved prediction accuracy.

8) Comparison: All the approaches contain similarities and differences. A synthesis of the different mechanisms used in each approach are summarized in the Table I.

V. Experimental Settings

A. Dataset

In order to illustrate our approach in a standard and reproductible context, we decide to use the dataset CIFAR-10 [27] dedicated to image classification. This dataset contains 60000 32×32 color images divided in 10 classes : airplanes, automobiles, birds, cats, deers, dogs, frogs, horses, ships and trucks. We have an initial division where the training dataset corresponds to 50,000 images, and the remaining 10,000 images correspond to the held-out dataset. From this initial training dataset, 90% will correspond to the retain dataset, and the other 10% corresponding to 5,000 images are the data to be forgotten (forget dataset). As for the held-out dataset, it will be divided into 80/20 distribution for the test and validation datasets respectively.

B. Experiment Objective

The focus of this article is about comparing the performance of different machine unlearning approaches. The comparison consider the runtime, the final outcomes of the model on the different datasets and its robustness. Our aim is to highlight the approaches offering the best trade-off.

C. Evaluation Metrics

To evaluate the performance of our models, we use common metrics typically employed in the training of neural networks, such as accuracy and loss. However, in this case, we pay attention not only to the training and validation datasets but

Approach	**Re-init. Weights**	**Re-init. Gradients**	**Fine Tuning on Valid**	**Fine Tuning on Forget**	**Fine Tuning on Retain**	**Competition Place**
Distillation	✓		✓		✓	6
Rotate	✓				✓	5
Pseudo				✓	✓	5
Pruning	✓				✓	4
Deviation	✓				✓	3
Gradient	✓	✓			✓	2
Divergence				✓	✓	1

TABLE I: Synthesis of the Different Mechanisms Used in Each Approach.

also to the test, retain, and forget datasets to observe their behavior during each epoch. In addition to these metrics, we analyze runtime efficiency and AUC (Area Under the Curve) scores of ROC (Receiver Operating Characteristic) Curve in the case of a membership inference attack (MIA), which will be explained below.

1) Accuracy on retain, forget and validation datasets: The accuracy measure the ability to the model to well predict. Our objective is double:

- Preserve (or increase) the accuracy for the validation and the retain datasets compare to the original model.
- Decrease the accuracy for the forget dataset compare to the original model. Although it should not be significantly lower or similar than the validation accuracy.

Denote that we consider the value on the different losses, but for the sake of brevity and clarity, in this article we will only display accuracies.

2) Run Time Efficiency: Run Time Efficiency is a measure of the computational resources and time required to perform a specific task, such as training, inference, or unlearning in machine learning models.In machine unlearning, run time efficiency is critical as it affects the practicality and scalability of the unlearning processes [1]. Efficient unlearning techniques ensure that data can be removed quickly and with minimal computational overhead, making it feasible to implement unlearning in real-world applications where timely compliance with data removal requests is essential. This also highlights that we want the unlearning to take much less time than retraining the model from scratch.

3) AUC Score of ROC Curve under MIA Attack: The Membership Inference Attack (MIA) process begins by grouping data from the test and forget datasets, followed by their comparison. It seeks to determine if a specific sample was part of the model's forget dataset, based on the loss assigned by the model. Losses for each sample are computed, and binary labels (1 for forget, 0 for test) are assigned. These losses and labels are then used to train a logistic regression model that learns to differentiate between the two datasets. Stratified cross-validation is used to train and evaluate the attack model, ensuring that each subset maintains the original proportion of test and forget samples. Finally, the attack model's accuracy is calculated for each subset, reflecting how well it distinguishes between forget and test samples.

By analysing the classification threshold of the attack model with obtain the AUC of ROC Curve of a MIA attack. The ROC Curve represents the true positive rate against the false positive rate at each classifier threshold setting.

An AUC close to 0.5 means that the attack model is not able to distinguish between the forget and the test datasets. Thus it will be difficult for an attacker to infer information on the unlearned dataset from the unlearned model.

VI. Results

The variability of the results is assessed across different splits between the Retain dataset and the Forget dataset, i.e., how each model behaves with different subsets of data. This process was repeated over 10 iterations.

A. Model Performance

During the forgetting process performed by each approach, an analysis of accuracy and loss was conducted at each epoch, with the aim of observing particularly the decrease in accuracy and the increase in loss during the early stages of forgetting, due to the techniques that directly affect the model's head or layers. Subsequently, the increase in accuracy and the decrease in loss were observed during the fine-tuning process performed by each approach. The difference will be especially noticeable in the final accuracy result.
At the top of Figure 12, we provide the boxplots of the accuracies of each approach on the retain and validation datasets, both with a random and a target forget sets, to allow comparison with the original model, i.e.,before the unlearning process. In addition to the approaches already studied and the original model, we will also provide the accuracies of a model we have termed “simple”, as its unlearning process involves only fine-tuning on the retain dataset without any additional processing, unlike the other approaches and of the original model. There, we do not represent gradient method, which is significantly worst than the other methods. Keeping it on would greatly reduce the readability of the figure. Useful unlearning is not supposed to degrade the model's performance, so the retain and validation accuracies should remain close to those of the original model.

At the bottom of Figure 12, we plot the ratio between the accuracies on the forget and validation sets. An effective unlearning approach should yield a ratio close to 1. As shown

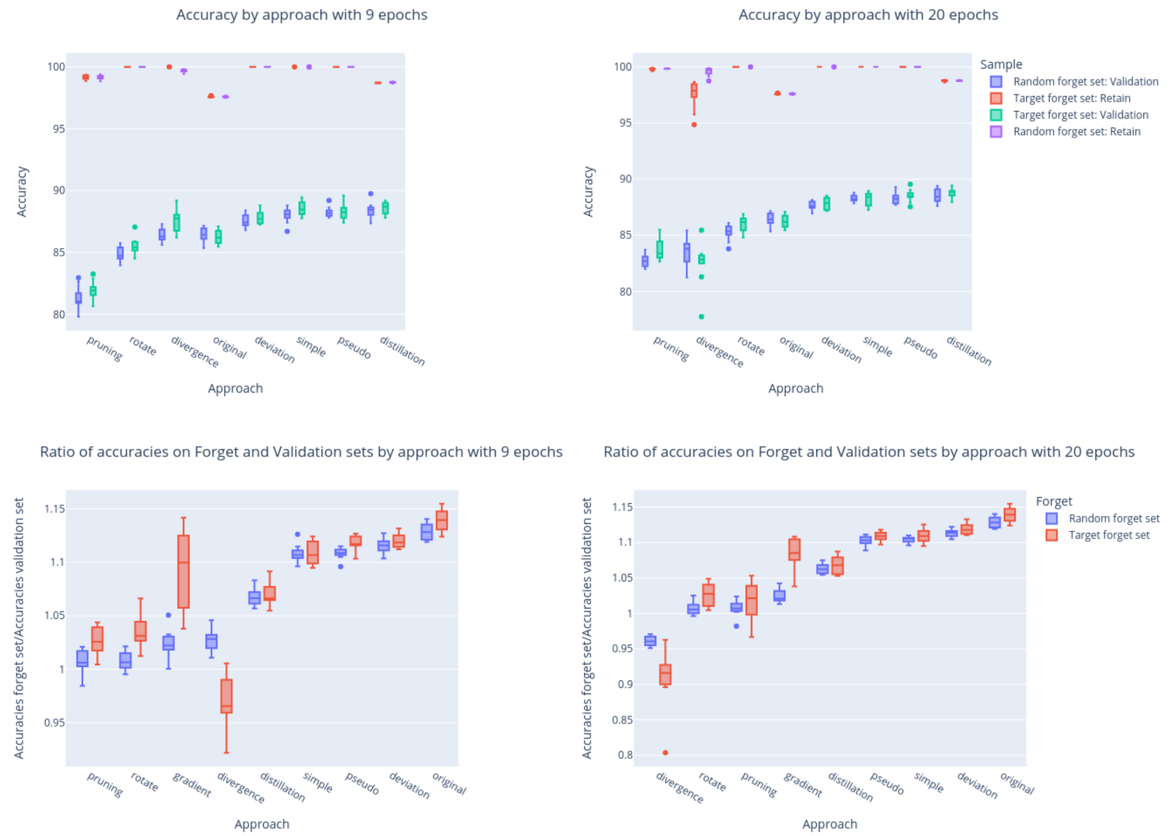

FIG. 12: (Top) Accuracies of Retain and Validation sets by approach and dataset. (Bottom) Ratio between accuracies on forget and validation sets by approach and dataset. Both respectively with 9 (Left) and 20 (Right) epochs.

in the figure, the original model displays differing accuracies between the validation and forget sets, which is expected, as no unlearning method has been applied and the forget dataset was randomly selected.

For the distillation, pseudo-labeling, simple, and deviation approaches, the accuracies between the forget and validation datasets are quite different, indicating incomplete unlearning. For all these approaches, considering 9 or 20 epochs lead to similar results. Gradient approach requires a greater number of epochs than the others unlearning approaches to attain a similar performance to that of the original model on the retain dataset. A plausible hypothesis is that the utilization of gradients for unlearning may be an effective strategy, yet it poses a challenge for the model to regain its original performance, potentially due to the complexity of the unlearning process. Rotate and pruning approaches allow to make accuracies of validation and forget datasets close. However, it degrades the performance on the validation datasets (pruning more than rotate). Considering more epochs allow a smaller degradation and reduce variability. Divergence is more dependent than the others approaches to the number of epochs. With 9 epochs, the accuracies on the validation datasets are a little stronger than the accuracies of the original model. The accuracies on the forget datasets are stronger but close to the accuracies on the validation datasets. With 20 epochs, the accuracies on the validation datasets are a little smaller than the accuracies of the original model (and with more variability). The accuracies on the forget datasets are smaller but close to the accuracies on the validation datasets. This let us think that with a good choice of the number of epoch, we should be able to have similar accuracies on the forget and validation datasets and accuracies on the validation dataset close to the accuracies of the original model.

Furthermore, we computed the approaches with a reduced number of epochs, specifically 5, and observed that the results exhibited negligible variation compared to those achieved at 9 epochs for most of the approaches, suggesting that the models had already converged to a stable performance at this earlier stage. This suggests that the models are able to learn and unlearn effectively within a relatively small number of epochs, and that further increases in the number of epochs may not yield significant improvements in performance. Among the approaches evaluated, Rotate, Pruning, and Divergence emerged as the best performing.

B. *Run time efficiency*

Figure 13 provides the run time of each approach for the ten repetitions with 9 epochs. First, all approaches exhibit a significant reduction in computational time, with none requiring more than half the time needed to re-train the original model, which takes around 41 minutes and 41 seconds. Furthermore, the simple, deviation and divergence models demonstrate the fastest unlearning times, a characteristic that underscores their robustness in terms of run time efficiency. Pseudo labeling takes more time and divergence is faster than the others unlearning approaches.

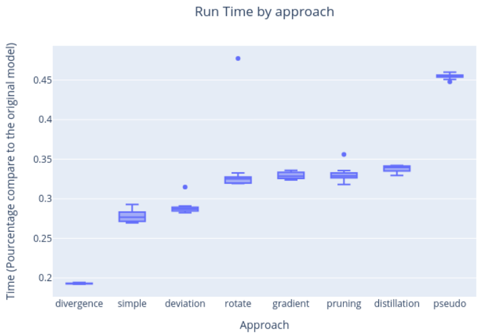

FIG. 13: Run Time by approach with 9 epochs, as a percentage of the training time of the original model.

C. MIA

Consistent with the previous metrics, we evaluated the robustness of the unlearning approaches against MIA Attacks considering the 10 experiment repetition. The results, presented in the left part of Figure 14, illustrates the variability in AUC across each approach, as well as the original model. It is noteworthy that an AUC closer to 0.5 indicates a more effective unlearning process, as the attack is unable to differentiate between samples from the forget dataset and those from the test dataset, suggesting that the forget data is indistinguishable from unseen data to the model. This evaluation provides a robust assessment of our approach's ability to protect sensitive information, and the results demonstrate its effectiveness in mitigating MIA Attacks. Some results of Figure 14 are coherent with Figure 12. All models demonstrate a reduction of MIA AUC Score compared to the original model. The MIA AUC Score of simple, deviation, pseudo and distillation is stronger than for the other unlearning approaches. Moreover, the MIA AUC Score of rotate, pruning, gradient and divergence is close to 0.5, the attack perform almost as well as a random attack. The attack on the pruning seems the closer to a random attack.

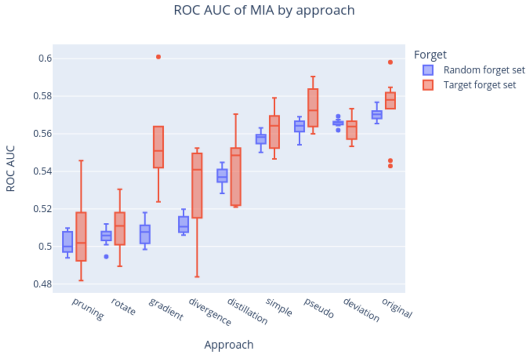

FIG. 14: AUC scores in case of MIA Attack by approach with 9 epochs with forget set randomly chosen and forget set targeted on red images.

D. Realistic forget dataset

To push our benchmark further we have compared the different approaches with a target forget dataset. We have selected 525 pictures having a lot of red, to be part of the forget dataset. To select these images, we have used the process proposed by [43]. The accuracies on the retain and the validation sets as well the ratio between the accuracies on the forget and the validation sets are given in Figure 12. In the majority of case, the range of the boxplots are larger than when the forget set is chosen randomly. Despite this, the overall behavior of the methods remains close to that observed in the random choice context. Methods like rotate and pruning whose forget and validation accuracies are close in the case of random forget set selection keep them close in the case of targeted forget set selection. For the sake of clarity, we have not shown the results with 5 epochs in the Figure 12. However, for the divergence method, with 5 epochs, the accuracies of the forget and validation sets are the closest, while maintaining a high level of performance on the retain set. On the right part of Figure 14, we see that the AUC with target forget dataset has more variability than random forget dataset. Besides, the MIA is almost random when rotate or pruning methods are used.

VII. Conclusion

Our study highlights the importance of striking a balance between performance, time, and defense against attacks in machine unlearning models. While achieving high performance is crucial, it must be accomplished within a reasonable time frame to maintain efficiency. Moreover, defense against attacks is vital to prevent malicious actors from manipulating the model's outputs or stealing sensitive data. Our results identify that rotate, pruning, and divergence approaches offer a promising balance between these three aspects, although the variability of the divergence method's results must be considered. These findings provide a foundation for future research and development in the field of data removal and machine learning.

We consider both a randomly selected forget dataset and a more realistic representation of real-world data privacy scenarios where the forget dataset is based on specific image characteristics, such as color. Identifying these features allows for more accurate data removal, as demonstrated in [21] for future researches.

Future research directions include exploring class removal and the use of alternative datasets, as well as developing metrics such as the Amnesis index [44] and MIA attack based on LiRA [23] to determine the most robust method for defending against attacks.

References

[1] T. Shaik, X. Tao, H. Xie, L. Li, X. Zhu, and Q. Li, "Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy," 2024. [Online]. Available: https://arxiv.org/abs/2305.06360

[2] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang, "Biometrics recognition using deep learning: A survey," 2021. [Online]. Available: https://arxiv.org/abs/1912.00271

[3] C. Ahmed, M. Umer, S. Liyakkathali, M. T. Jilani, and J. Zhou, *Machine Learning for CPS Security: Applications, Challenges and Recommendations*, 12 2020, pp. 397–421.

[4] K. Z. Liu, "Machine unlearning," https://ai.stanford.edu/~kzliu/blog/unlearning, 2020, accessed: 2023-02-20.

[5] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," 2020. [Online]. Available: https://arxiv.org/abs/1911.04933

[6] B. Buet, G. P. Leonardi, and S. Masnou, "Weak and approximate curvatures of a measure: a varifold perspective," 2020. [Online]. Available: https://arxiv.org/abs/1904.05930

[7] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," 2020. [Online]. Available: https://arxiv.org/abs/1912.03817

[8] D. M. Sommer, L. Song, S. Wagh, and P. Mittal, "Towards probabilistic verification of machine unlearning," 2020. [Online]. Available: https://arxiv.org/abs/2003.04247

[9] N. Li, C. Zhou, Y. Gao, H. Chen, A. Fu, Z. Zhang, and Y. Shui, "Machine unlearning: Taxonomy, metrics, applications, challenges, and prospects," 2024. [Online]. Available: https://arxiv.org/abs/2403.08254

[10] L. Riecke, "Unmasking the Term 'Dual Use' in EU Spyware Export Control," *European Journal of International Law*, vol. 34, no. 3, pp. 697–720, 09 2023. [Online]. Available: https://doi.org/10.1093/ejil/chad039

[11] J. Xu, Z. Wu, C. Wang, and X. Jia, "Machine unlearning: Solutions and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 3, p. 2150–2168, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1109/TETCI.2024.3379240

[12] Y. Xu, "Machine unlearning for traditional models and large language models: A short survey," 2024. [Online]. Available: https://arxiv.org/abs/2404.01206

[13] Y. Qu, X. Yuan, M. Ding, W. Ni, T. Rakotoarivelo, and D. Smith, "Learn to unlearn: A survey on machine unlearning," 2023. [Online]. Available: https://arxiv.org/abs/2305.07512

[14] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 463–480.

[15] C. Chen, F. Sun, M. Zhang, and B. Ding, "Recommendation unlearning," 2022. [Online]. Available: https://arxiv.org/abs/2201.06820

[16] M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou, "Towards unbounded machine unlearning," 2023. [Online]. Available: https://arxiv.org/abs/2302.09880

[17] L. Graves, V. Nagisetty, and V. Ganesh, "Amnesiac machine learning," 2020. [Online]. Available: https://arxiv.org/abs/2010.10981

[18] T. Baumhauer, P. Schöttle, and M. Zeppelzauer, "Machine unlearning: Linear filtration for logit-based classifiers," 2020. [Online]. Available: https://arxiv.org/abs/2002.02730

[19] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," 2022. [Online]. Available: https://arxiv.org/abs/2209.02299

[20] Triantafillou, Pedregosa, Hayes, Kairouz, Guyon, Kurmanji, Dziugaite, Triantafillou, Zhao, S. Hosoya, J. C. S. J. Junior, Dumoulin, Mitliagkas, Escalera, D. Wan, Demkin, and Reade, "Neurips 2023 - machine unlearning," 2023. [Online]. Available: https://kaggle.com/competitions/neurips-2023-machine-unlearning

[21] Y. Zhang, S. Yao, S. Shen, S. Xu, T. Yang, B. Li, X. Zhang, Z. Shao, and Z. Gu, "The secret revealer: Generative model-inversion attacks against deep neural networks," 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/Zhang_The_Secret_Revealer_Generative_Model-Inversion_Attacks_Against_Deep_Neural_Networks_CVPR_2020_paper.pdf

[22] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2017. [Online]. Available: https://arxiv.org/abs/1610.05820

[23] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, "Membership inference attacks from first principles," 2022. [Online]. Available: https://arxiv.org/abs/2112.03570

[24] S. Jiang, Y. Luo, S. Zheng, Y. Yu, Z. Xu, Y. Tan, and J. Zhao, "Membership inference attacks against recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9833649

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[26] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1998.

[27] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

[28] E. Karypidis, V. Perifanis, C. C. Nikolaidis, N. Komodakis, and P. Efraimidis. (2023) 6th place solution. [On-

line]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458740

[29] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation : A survey,” 2021. [Online]. Available: https://arxiv.org/abs/2006.05525

[30] Encord. (2023) Kl divergence in machine learning. [Online]. Available: https://encord.com/blog/kl-divergence-in-machine-learning/

[31] A. Mao, M. Mohri, and Y. Zhong, “Cross-entropy loss functions: Theoretical analysis and applications,” pp. 23 803–23 828, 2023. [Online]. Available: https://proceedings.mlr.press/v202/mao23b.html

[32] Marvelworkd and Toshi-k. (2023) 5th place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458531

[33] S. Oleszko. (2023) 4th place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459148

[34] J. Jia, J. Liu, P. Ram, Y. Yao, G. Liu, Y. Liu, P. Sharma, and S. Liu, “Model sparsity can simplify machine unlearning,” 2024. [Online]. Available: https://arxiv.org/pdf/2304.04934

[35] S. Achour. (2023) 3rd place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459334

[36] “Backpropagation and stochastic gradient descent method,” *Neurocomputing*. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/092523129390006O

[37] D. Lee, J. Bae, and J. Kim. (2023) 2nd place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459200

[38] N. Lee, T. Ajanthan, and P. H. Torr, “Snip: Single-shot network pruning based on connection sensitivity,” in *International Conference on Learning Representations*, 2019.

[39] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *arXiv preprint arXiv:1502.01852*, 2015.

[40] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1601.06759*, 2016. [Online]. Available: https://arxiv.org/abs/1601.06759v3

[41] Fanchuan. (2023) 1st place solution. [Online]. Available: https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458721

[42] T. Wang and P. Isola, “Understanding the behaviour of contrastive loss,” 2021. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Wang_Understanding_the_Behaviour_of_Contrastive_Loss_CVPR_2021_paper.html

[43] K. Bhanot, “Color identification in images,” 2018. [Online]. Available: https://towardsdatascience.com/color-identification-in-images-machine-learning-application-b26e770

[44] M. M. M. K. Vikram S Chundawat, Ayush K Tarun, “Zero-shot machine unlearning,” 2022. [Online]. Available: https://arxiv.org/abs/2201.05629