SFTC: Machine Unlearning via Selective Fine-tuning and Targeted Confusion

Vasileios Perifanis[*]
vperifan@ee.duth.gr
Democritus University of Thrace
Greece

Efstathios Karypidis[*]
e.karypidis@athenarc.gr
National Technical University of Athens,
Archimedes/Athena RC
Greece

Nikos Komodakis
komod@csd.uoc.gr
University of Crete, IACM-Forth, Archimedes/Athena RC
Greece

Pavlos S. Efraimidis
pefraimi@ee.duth.gr
Democritus University of Thrace, Athena RC
Greece

ABSTRACT

As the importance of data privacy escalates in the modern digital era, machine learning service operators face challenges posed by the stringent privacy regulations, such as the GDPR. To cope with these challenges, the concept of machine unlearning emerges as a key solution that meets data removal requirements, while maintaining trust and transparency, thereby reducing the risk of data breaches. In this work, we present a **S**elective **F**ine-tuning and **T**argeted **C**onfusion (SFTC) algorithm for machine unlearning. SFTC simultaneously performs fine-tuning on the remaining data and selectively confuses the original model by following the distribution of a biased random generator, effectively leading the forget samples' output space to be indistinguishable from that of the original test samples. Our algorithm is evaluated on three diverse datasets for image classification and its unlearning performance is compared against six state-of-the-art unlearning algorithms. The results show that SFTC preserves a model's original accuracy while effectively inducing forgetting on the requested data samples.

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**; • **Security and privacy → Human and societal aspects of security and privacy**; *Digital rights management*.

KEYWORDS

Machine Unlearning, Deep Learning, Data Privacy, Machine Learning Security and Privacy

ACM Reference Format:
Vasileios Perifanis, Efstathios Karypidis, Nikos Komodakis, and Pavlos S. Efraimidis. 2024. SFTC: Machine Unlearning via Selective Fine-tuning and Targeted Confusion. In *European Interdisciplinary Cybersecurity Conference (EICC 2024), June 05–06, 2024, Xanthi, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3655693.3655697

EICC 2024, June 05–06, 2024, Xanthi, Greece
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1651-5/24/06
https://doi.org/10.1145/3655693.3655697

1 INTRODUCTION

Machine learning models, notably those like GPT and Dall-E, have revolutionized everyday tasks in various sectors [17]. Typically, these models are developed by collecting user data in a datacenter, followed by processing through machine learning pipelines [12]. However, privacy regulations like the GDPR [23] require that service providers delete users' data upon request. In addition, from a security perspective, removing the influence of samples from machine learning models reduces model errors and the risk of adversarial attacks [24] like membership inference [19] and model inversion [5], which compromise service confidentiality.

To meet users' requests and comply with regulations, service providers should erase not only the associated users' data but also modify their deployed models to reflect this deletion [16, 24, 25]. The process of making trained learning models forget in a time-efficient manner is referred to as *machine unlearning* [2].

The most straightforward approach for unlearning is to retrain the model from scratch without the forget set, a process called *exact unlearning* [22]. However, this method is impractical due to its significant computational costs and time consumption. Furthermore, as data deletion requests can occur arbitrarily, retraining for each request is not feasible. Consequently, *approximate unlearning* [4, 7, 14] emerged as a key solution that modifies the original model efficiently while maintaining its predictive accuracy.

In this work, we introduce an unlearning algorithm that adjusts the original model trained on the complete dataset. Our algorithm, Selective Fine-tuning and Targeted Confusion (SFTC) utilizes a teacher-student approach and ensures that the process does not exceed 15% of the original training duration. Specifically, it fine-tunes the original model on the retain set (remaining data), while confusing it on the forget set using a biased random output generator.

Our main contributions are summarized as follows:

- We propose SFTC, a novel machine unlearning algorithm that refines the original model on the retain set, while distancing its predictions on the forget set from those of the original model, using a biased random generator.

[*] Both authors contributed equally to this research.

- We introduce a new forget set benchmark on the FER-2013 dataset, which includes samples from two classes and incorporates in-context information. Specifically, the forget set consists of images from minors, providing a distinct context for evaluating the unlearning process.
- We evaluate SFTC on a diverse set of datasets and compare it against six unlearning algorithms. Our results suggest that SFTC effectively induces forgetting on the requested data.

The remainder is structured as follows. Section 2 outlines the concept of machine unlearning. Section 3 summarizes the related work. Section 4 introduces the SFTC algorithm. Section 5 presents the experimental results and compares SFTC with state-of-the-art unlearning algorithms. Finally, Section 6 concludes our work and discusses future directions.

2 PROBLEM DEFINITION

In this section, we formally define the problem of machine unlearning given a trained model, the original dataset and the specified data subset that need to be forgotten.

Machine Learning. Let a dataset $\mathcal{D}$, represented as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with N samples. Each sample consists of a d-dimensional feature vector $x_i \in \mathbb{R}^d$ and its corresponding label $y_i \in \{0, ..., C-1\}$ with C being the number of classes. A machine learning algorithm $f_\theta(\cdot)$, where θ denotes the model parameters, is applied to $\mathcal{D}$ in a supervised manner. The objective is to choose $f_\theta(\cdot)$ such that $\hat{y}_i = f_\theta(x_i)$ approximates the true label y_i. To achieve this objective, we aim to minimize a loss $\mathcal{L}_D = \frac{1}{N}\sum_{i=1}^{N} l\left(f_\theta\left(x_i\right), y_i\right)$, where l is a loss function, such as cross-entropy. In this work, we are interested in deep learning models, i.e., $f_\theta(\cdot)$ characterizes a neural network model M with multiple operational layers, defined by its weights θ.

Machine Unlearning. Let the requested set of samples that needs to be forgotten is represented as $\mathcal{D}_f \subset \mathcal{D}$, which corresponds to a subset of the original dataset. The retain dataset (remaining data), $\mathcal{D}_r$, is obtained by excluding $\mathcal{D}_f$ from $\mathcal{D}$, i.e., $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. We assume that $\mathcal{D}_f$ comprises a random subset from multiple classes. Given the original model M with weights θ trained on $\mathcal{D}$, along with the retain set $\mathcal{D}_r$ and the forget set $\mathcal{D}_f$, the goal is to apply an unlearning algorithm $U(\cdot)$. The unlearning process modifies the original model's weights θ into new weights θ_u, resulting in a new model M_u. The goal for M_u is to unlearn $\mathcal{D}_f$, while maintaining high utility (e.g., high accuracy), similar to the original model.

Fig. 1 illustrates the process of machine unlearning from a service provider's point of view. Initially, users (data owners) share their data with the provider. After collection, the dataset is employed to train a machine learning model, which can generate useful predictions for customers (model consumers), who can be either data owners or other external users. Following the service deployment, a subset of data owners request the deletion of their associated information. In response, the service provider should not only remove the data from their local databases but also make the previously trained model unlearn these data. This step is crucial to comply with the "right to be forgotten" directive of regulations such as the GDPR, which also fulfills users' desiderata.

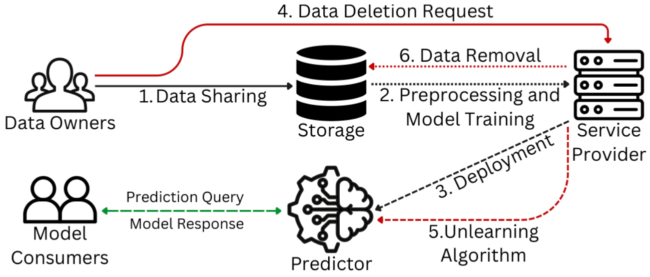


Figure 1: Overview of Machine Unlearning.

3 RELATED WORK

The concept of machine unlearning, introduced by Cao and Yang [2], focuses on efficient, exact data removal using summation methods based on statistical query learning. While efficient, this unlearning algorithm is limited to simple algorithms such as Naive Bayes and cannot scale to more complex models like neural networks. Bourtoule et al. developed SISA [1], which partitions the original training dataset into disjoint shards, with each shard having its own isolated sub-model. When an unlearning request arrives, only the sub-models with that sample are being retrained. However, it necessitates initial adjustments in the training phase and may not be applicable in many scenarios due to the partitioning strategy. Ginart et al. [6] proposed a method for approximate unlearning, focused on k-means clustering, through quantization and data partitioning. However, it is effective in models with a limited number of parameters, limiting its usability for neural networks.

One of the earliest works for unlearning in deep neural networks, NegGrad [8], involves adjusting the original model parameters by using gradient ascent for the forget set. Choi [3] enhanced the NegGrad approach by including an additional fine-tuning loss term and introduced two real-world image datasets for evaluating machine unlearning algorithms. Graves et al. [10] proposed amnesiac unlearning where the forget samples are assigned a random label and fine-tuning is performed on the concatenation of the retain and forget sets after re-labeling. Goel et al. [7] proposed two methods for unlearning, CF-k and EU-k forgetting. The former involves freezing the first k layers and performing fine-tuning using the rest model layers on the retain set and the latter starts by randomly initializing the rest layers before fine-tuning.

Our proposed algorithm builds upon Bad-Teaching [4] and is closely related to the SCRUB [14] algorithm. Bad-Teaching uses a two-teacher approach, where the retain and forget samples are relabeled to 0 and 1, respectively. Then, the student model is trained to generate a similar behavior to that of the original model on the retain set and a completely random model on the forget set. SCRUB combines fine-tuning on the retain set and minimizes the divergence between the student and original models on the retain set, while maximizing it on the forget set. Both Bad-Teaching and SCRUB try to confuse the model on the forget set with random predictions or by following a different direction. In this work, we argue that the above two methods might affect a larger number of samples than intended, particularly those in the retain or the original test set. To address this issue, we propose a method for targeted confusion on the forget set, using a controlled biased output generator.

Evaluating Machine Unlearning. One of the most controversial aspects of machine unlearning is how to evaluate an unlearned model [7, 14, 21]. An ideal unlearning algorithm produces a model with high-quality predictions, similar to the original model, while ensuring that data are effectively forgotten. While it is straightforward to compare the unlearned model's output with the original model, proving effective unlearning is more complex. Many studies assess this by comparing the unlearned model to one retrained from scratch on the available data [4, 14]. However, this method is often impractical and fails to address the stochastic variability in machine learning, implying that indistinguishability between an unlearned and a retrain-from-scratch model is not a reliable unlearning indicator [7, 21]. In this work, we evaluate our unlearning algorithm with both approaches to ensure comprehensive assessment.

4 SELECTIVE FINE-TUNING AND TARGETED CONFUSION (SFTC)

In this section, we introduce SFTC, a refined unlearning algorithm based on [4]. SFTC fine-tunes the original model on the retain set $\mathcal{D}_r$, follows the original model M trained on the entire dataset $\mathcal{D}$ in terms of output distribution for $\mathcal{D}_r$ and selectively tries to diverge its predictions from M on the forget set $\mathcal{D}_f$ by following a biased random distribution generator M_b.

In SFTC, the original model M, with weights θ, serves as the teacher model for $\mathcal{D}_r$ and a random generator model M_b acts as the teacher for $\mathcal{D}_f$. Both models process an input sample x and produce a logit $z^{(x)}$, which is then transformed into a probability distribution via softmax activation. Our objective is to train a student model M_u initialized with θ and yield weights θ_u, such that M_u selectively forgets $\mathcal{D}_f$ while retaining knowledge from $\mathcal{D}_r$.

We begin our approach by augmenting the retain and forget set with a pseudo-label $b \in \{0, 1\}$ assignment. Specifically, samples from $\mathcal{D}_r$ are assigned $b = 0$ and those from $\mathcal{D}_f$ with $b = 1$. This results in augmented with pseudo-labels sets $\mathcal{D}_r^{'}$ and $\mathcal{D}_f^{'}$. These sets are then combined into a unified unlearning dataset $\mathcal{D}_u = \mathcal{D}_r^{'} \cup \mathcal{D}_f^{'}$. The idea for assigning pseudo-labels was influenced by the Bad-Teaching unlearning approach [4], where the authors replaced the actual labels with pseudo-labels. The unlearning dataset $\mathcal{D}_u$ is subsequently shuffled and partitioned into batches for training.

Selective Fine-Tuning. To fulfill the predictive utility preservation requirement, the SFTC algorithm first performs a fine-tuning operation on $\mathcal{D}_r$. During training, the algorithm selects the samples that correspond to pseudo-label $b = 0$ and minimize the cross-entropy loss, defined as:

$$\mathcal{CE}_r = -\sum_{i=1}^{N} \sum_{k=1}^{C} y_{ik} \log\left(\hat{y}_{ik}\right), \tag{1}$$

where N is the number of samples, C is the number of classes, y_{ik} is 1 if the ground truth class of the ith sample is k and $\hat{y}_{ik}$ is the predicted probability of the ith sample belonging to class k.

Targeted Confusion. Besides fine-tuning, SFTC uses the original model M to guide the student model M_u towards a similar output distribution on $\mathcal{D}_r$ and the random generator model M_b towards differing output distribution on $\mathcal{D}_f$. Similar to the Bad-Teaching approach [4], we optimize the Kullback-Leibler (KL) divergence:

$$\begin{aligned} \mathcal{KL} &= (1-b)\, D_{KL}\left(M(x)\,||M_u(x)\right) + (b) D_{KL}\left(M_b(x)\,||M_u(x)\right) \\ &= (1-b) \sum_{x \in \mathcal{D}_u} M(x) \log\left(\frac{M(x)}{M_u(x)}\right) + (b) \sum_{x \in \mathcal{D}_f} M_b(x) \log\left(\frac{M_b(x)}{M_u(x)}\right), \end{aligned} \tag{2}$$

where b is the assigned pseudo-label, x is an input sample and $M_{(\cdot)}(x)$ denotes the output of model $M_{(\cdot)}$ for sample x after applying softmax. Note that the outputs of $M_{(\cdot)}$ can be scaled according to a temperature parameter τ. By default, $\tau = 1$.

Optimizing the loss in Eq. 2 allows the student model to follow both teachers with respect to their output distribution on $\mathcal{D}_r$ and $\mathcal{D}_f$, respectively. However, if we set a completely random model M_b as in [4] and we follow this random generator on the forget set, M_u may be confused on a larger fraction of samples. For instance, suppose that we have a specific sample that needs to be forgotten. This sample's features are very similar to that of a random sample's features belonging to the retain set. In this sense, if we make a random prediction on the forget sample, we will influence the model in making errors on similar retain samples. Consequently, this will lead the model to be confused in a larger fraction of samples than expected, which will further lead to inconsistencies.

To address this issue and effectively confuse the student model, we propose a targeted approach using a biased random output generator. The generator tailors predictions for the forget set, which can vary from being completely random to being biased towards the correct class. The degree of bias is governed by a scalar c, allowing for controlling the confusion during unlearning.

More precisely, the generator creates a random output distribution for each input data, where the distribution's size is determined by the number of classes present in the original dataset. To generate the output, we sample from the normal distribution. Initially, the outputs are completely random. We then employ the scalar c to infuse a specific degree of confusion, adjusting the initial randomness in a targeted manner. When $c = 1$, the outputs remain entirely random, similar to the Bad-Teaching approach [4]. In contrast, with $c = 0$, the output is carefully adjusted to align with the correct label for each sample. This adjustment involves adding a random number to the corresponding correct class index in the output distribution. By default, SFTC uses $c = 0$ and intuitively, the model retains the correct labels but its confidence in the forget set samples is reduced, leading to the desired targeted confusion.

Our method is flexible, allowing any level of confusion between 0 and 1, where higher values of c result in greater confusion. This concept is similar to [10], where the target labels in the forget set are assigned randomly. To achieve this, we select the batch indices corresponding to forget samples, i.e., the samples with pseudo-label $b = 1$. Then, we get the number of samples to change their associated label by multiplying the number of forget samples in the batch with the scalar c and selecting as many samples uniformly at random. The selected samples are assigned a new label from $[0, C-1]$ and the generator is biased towards these random labels.

Putting it all together, SFTC optimizes both losses (Eq. 1 and 2) to effectively induce forgetting on $\mathcal{D}_f$ governed by the confusion fraction c while retaining high accuracy on $\mathcal{D}_r$:

$$\mathcal{L}_{SFTC} = \mathcal{CE}_r + \mathcal{KL} \tag{3}$$

5 EXPERIMENTS

In this section, we outline the experimental setup and assess the performance of different unlearning algorithms. [1]

5.1 Datasets

We evaluate SFTC using three image datasets. Specifically, we consider the **CIFAR-10** dataset [13], which consists of ten balanced classes with 5,000 images each. The forget set represents 10% of each class (500 images each) and is provided by Google.[2] The second dataset is **MUFAC** [3], comprising facial images for age group prediction. The dataset is imbalanced and the forget set mirrors the original imbalance. Finally, we present a new forget set benchmark for evaluating unlearning on the **FER-2013** dataset [9], which consists of facial expressions across seven imbalanced classes. For the forget set split, we consider a scenario aligning with a (conceptual) new legislative requirement, where images tagged with fear or sadness from minors should be removed from learning models. In this scenario, the forget set is limited to a subset of only two classes. Fig. 2 presents a sample from the facial images belonging to the FER-2013 forget set. Fig. 3 illustrates the distribution of samples per class across training, validation and test sets for each dataset as well as the distribution per class in the forget sets.

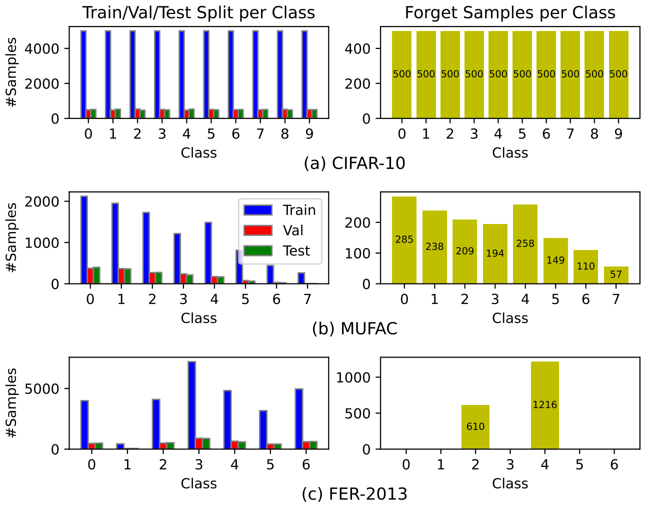
5.2 Experimental Setup

To assess the unlearning performance we first train the ResNet-18 [11] and EfficientNet-B0 [20] models on the original datasets. The former architecture has been thoroughly assessed in machine unlearning literature [4, 7, 8, 14] and the latter is considered as a more complex and lightweight architecture. For model training, we use the Adam optimizer with an initial learning rate of 10^{-3} and the cosine annealing scheduler for 30 epochs with a batch size of 64. All experiments were conducted five times with different initialization seeds on NVIDIA RTX 3060 GPU-equipped workstation running Ubuntu 20.04 and PyTorch 2.0.1.

5.3 Unlearning Algorithms

We compare our proposed SFTC unlearning algorithm against the following baselines and state-of-the-art approaches. **Fine-Tuning** (FT) is the simplest baseline, where we begin from the original model and fine-tune it on $\mathcal{D}_r$ for a limited number of epochs. **Neg-Grad+** (NG+) [3, 8] combines fine-tuning on $\mathcal{D}_r$ and maximizing the error on $\mathcal{D}_f$. In **CF-k** and **EU-k Forgetting** [7] the first k layers are frozen and only the last layers are being fine-tuned, where the CF-k approach begins from the original weights and EU-k randomly initializes the rest layers. **Bad-Teaching** (BD) [4] involves optimizing the KL loss between the student and the original model on $\mathcal{D}_r$ and the KL loss between the student and a randomly initialized model on $\mathcal{D}_f$, similar to Eq. 2. **SCRUB** [14] performs fine-tuning on $\mathcal{D}_r$ (Eq. 1), minimizing the KL loss between the student model and the original model on $\mathcal{D}_r$ and maximizing the KL loss on $\mathcal{D}_f$. **Retrain** (RT) represents the ideal case, where the model is trained from scratch on $\mathcal{D}_r$.

Figure 2: FER-2013 Forget Samples.



(a) CIFAR-10

(b) MUFAC

(c) FER-2013

Figure 3: Dataset Distributions.

5.4 Evaluation Metrics

Unlearning Accuracy. To evaluate the unlearning accuracy, we assess the predictive performance of the unlearned model M_u on both the forget set $\mathcal{D}_f$ and test set $\mathcal{D}_t$ against a retrain-from-scratch oracle M_R. The accuracy error between M_u and M_R is calculated using the Symmetric Absolute Percentage Error (SAPE) [15]:

$$\text{SAPE}\,(a, b) = \frac{|b - a|}{b + a}. \tag{4}$$

Specifically, we compute the following metrics:

$$\text{AccErr}_{\{M_u, M_R\}} = \text{SAPE}\left(\text{Acc}_{M_R}^{(\mathcal{D}_t)}, \text{Acc}_{M_u}^{(\mathcal{D}_t)}\right), \tag{5}$$

$$\text{AccDis} = \text{SAPE}\left(\text{Acc}_{M_R}^{(\mathcal{D}_f)}, \text{Acc}_{M_u}^{(\mathcal{D}_f)}\right). \tag{6}$$

However, since obtaining the M_R in real-world scenarios is often impractical, we also compare the accuracy of M_u on $\mathcal{D}_t$ relative to the original model M:

$$\text{AccErr}_{\{M_u, M\}} = \text{SAPE}\left(\text{Acc}_{M}^{(\mathcal{D}_t)}, \text{Acc}_{M_u}^{(\mathcal{D}_t)}\right). \tag{7}$$

In all of the above accuracy error metrics, lower values indicate more successful unlearning. Specifically, Eq. 5 shows the unlearning *effectiveness*, Eq. 6 the unlearning *certifiability* (similarity in performance between M_u and M_R) and Eq. 7 evaluates post-unlearning *robustness* of M_u compared to M in terms of predictive accuracy.

Distinguishability from Original Model. Based on related literature [4, 8, 16, 24], the unlearning algorithm should produce a model M_u that is similar to a retrain-from-scratch oracle M_R. Nevertheless, as already stated, having access to the M_R is impractical. Hence, we

[1] The code is available at https://github.com/vperifan/SFTC-Unlearn.
[2] https://github.com/unlearning-challenge/starting-kit.

Table 1: Unlearning Algorithms Comparison.

		ResNet-18					EfficientNet-B0				
Data	Method	AccErr$_{\{M_u,M_R\}}$(↓)	AccDis(↓)	AccErr$_{\{M_u,M\}}$(↓)	JS(↑)	MIA(↑)	AccErr$_{\{M_u,M_R\}}$(↓)	AccDis(↓)	AccErr$_{\{M_u,M\}}$(↓)	JS(↑)	MIA(↑)
CIFAR-10	ORI	0.0074	0.0490	0.0	0.0	0.2700	0.0028	0.0498	0.0	0.0	0.3568
	RT	0.0	0.0	0.0074	0.3034	0.4630	0.0	0.0	0.0022	∞	0.4680
	FT	0.0277	$\underline{0.0077}$	0.0343	0.3976	0.4830	0.0169	0.0097	0.0199	∞	0.4987
	NG+	0.0206	0.0265	0.0272	0.1589	0.3513	0.0147	0.0195	0.0178	∞	0.4031
	CF-5	0.0278	0.0099	0.0344	0.4208	0.4863	0.0172	**0.0030**	0.0203	∞	0.4919
	EU-5	0.0448	0.0415	0.0513	0.6894	0.5188	0.0293	0.0199	0.0324	∞	0.4963
	SCRUB	0.0349	0.0139	0.0489	0.3164	0.5374	0.0234	0.0086	0.0265	∞	0.4898
	BD	**0.0089**	0.0092	**0.0155**	**0.7294**	$\underline{0.6085}$	$\underline{0.0109}$	0.0064	$\underline{0.0140}$	∞	$\underline{0.5332}$
	SFTC	$\underline{0.0096}$	**0.0054**	$\underline{0.0162}$	$\underline{0.7076}$	**0.7049**	**0.0091**	$\underline{0.0057}$	**0.0121**	∞	**0.5473**
MUFAC	ORI	0.0104	0.3455	0.0	0.0	0.2488	0.0188	0.2818	0.0	0.0	0.4359
	RT	0.0	0.0	0.0105	1.0767	0.6589	0.0	0.0	0.0068	0.3581	0.5426
	FT	0.0374	0.0287	0.0456	0.9418	0.6145	**0.0124**	0.0517	**0.0116**	0.3014	0.5811
	NG+	0.0322	0.1255	0.0391	0.4717	0.4248	0.0328	0.0203	0.0474	0.3157	0.5144
	CF-5	0.0293	0.0491	0.0381	0.8968	0.6415	0.0151	0.0883	0.0146	0.2473	0.5162
	EU-5	0.0330	0.0319	0.0388	0.9853	0.5538	0.0325	$\underline{0.0187}$	0.0463	0.3161	0.5832
	SCRUB	**0.0117**	0.0292	**0.0302**	1.0059	0.5931	$\underline{0.0134}$	0.0414	0.0219	0.3304	0.4924
	BD	0.0282	**0.0258**	0.0362	$\underline{1.0232}$	$\underline{0.7364}$	0.0169	0.0209	0.0195	**0.3729**	**0.7853**
	SFTC	$\underline{0.0271}$	$\underline{0.0277}$	$\underline{0.0339}$	**1.1057**	**0.7541**	0.0164	**0.0108**	$\underline{0.0138}$	$\underline{0.3529}$	$\underline{0.7724}$
FER-2013	ORI	0.0083	0.6495	0.0	0.0	0.1197	0.0039	0.6361	0.0	0.0	0.1197
	RT	0.0	0.0	0.0083	3.5581	0.7230	0.0	0.0	0.0389	2.4005	0.6340
	FT	0.0673	0.0462	0.0295	2.2371	0.6952	**0.0377**	0.0659	$\underline{0.0091}$	1.2943	**0.6965**
	NG+	0.0884	0.2292	0.0575	2.4408	0.5579	0.0788	0.1195	0.0431	∞	0.5591
	CF-5	0.0697	0.0636	**0.0169**	2.1782	0.7005	0.0457	0.0631	0.0121	1.2004	0.6167
	EU-5	0.0667	0.1543	0.0381	$\underline{2.5051}$	0.6056	0.0462	0.1018	**0.0085**	∞	0.6435
	SCRUB	0.0938	0.0725	0.0912	2.0631	0.5321	0.0949	0.3355	0.0488	∞	0.5591
	BD	$\underline{0.0655}$	**0.0404**	0.0287	2.3148	$\underline{0.7017}$	0.0491	$\underline{0.0568}$	0.0127	1.6802	0.6278
	SFTC	**0.0649**	$\underline{0.0436}$	$\underline{0.0281}$	**2.6859**	**0.7061**	$\underline{0.0438}$	**0.0537**	0.0162	1.7451	$\underline{0.6543}$

measure how similar M_u behaves on $\mathcal{D}_f$ compared to the original model M, based on the Jensen-Shannon (JS) divergence:

$$\text{JS}\left(P_{\mathcal{D}_f}||Q_{\mathcal{D}_f}\right) = \frac{1}{2}D_{KL}\left(P_{\mathcal{D}_f}||R\right) + \frac{1}{2}D_{KL}\left(Q_{\mathcal{D}_f}||R\right), \quad (8)$$

where $R = \frac{1}{2}\left(P_{\mathcal{D}_f} + Q_{\mathcal{D}_f}\right)$ is the mean distribution and $P_{\mathcal{D}_f}, Q_{\mathcal{D}_f}$ are the output probability distributions of M_u and M over $\mathcal{D}_f$, respectively. We choose JS over KL divergence since JS provides a symmetric and smoothed measure of the difference between two probability distributions. Intuitively, post-unlearning, the model should treat the samples of $\mathcal{D}_f$ as unseen, similar to an independent test set. In this context, the outputs of M_u and M should be *distinguishable*. A higher value of JS indicates a greater deviation from the original, suggesting effective unlearning.

Verifiability and Privacy. To asses the verifiability/privacy aspect of machine unlearning, we perform a MIA [19] against M_u. We construct a balanced dataset by sampling an equal number of instances from both $\mathcal{D}_r$ and $\mathcal{D}_t$ (with equivalent distributions) to train a MIA predictor. Specifically, we utilize the CatBoost classifier [18] as the attacker model using as input features the logit vectors produced from M_u. CatBoost has been empirically proven to significantly outperform other models like Logistic Regression and Support Vector Machines with respect to MIA. The model is then applied on $\mathcal{D}_f$ to determine how many samples are correctly identified as non training members:

$$\text{MIA} = \frac{TN}{|\mathcal{D}_f|}. \quad (9)$$

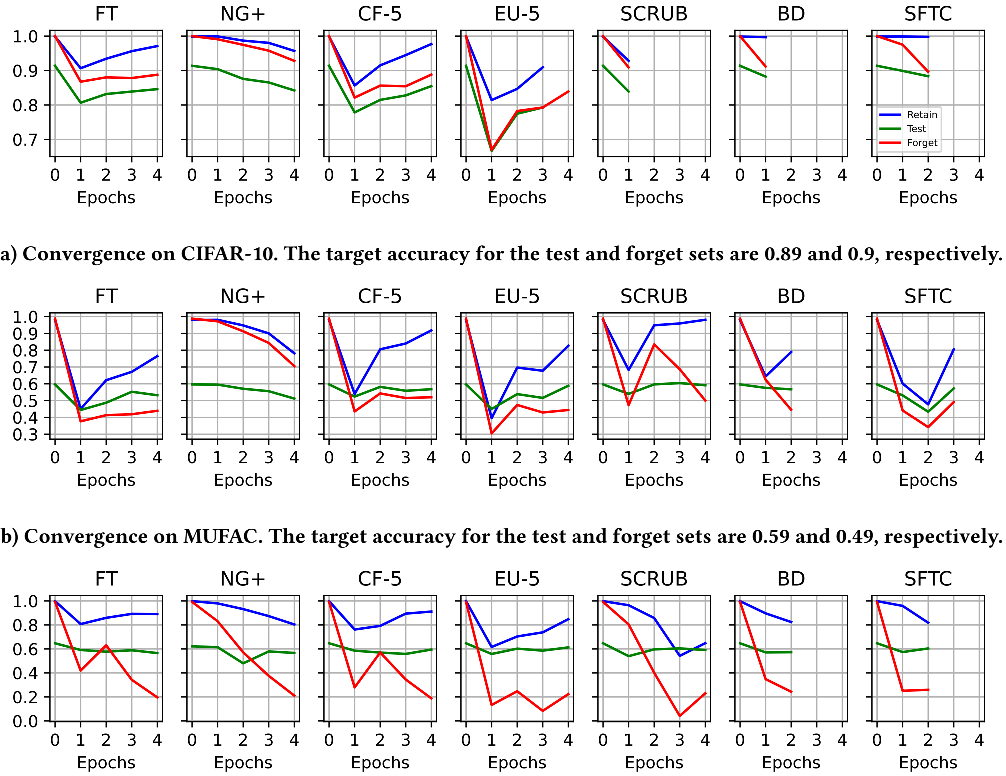
Higher values of this metric indicate higher *privacy* preservation for the forget samples and unlearning *verification* [24], i.e., the model's behavior on $\mathcal{D}_f$ is similar to that of unseen samples.

5.5 Results

To provide comprehensive results, we conducted a grid search across learning rates in $\{5 \times 10^{-3}, 4 \times 10^{-3}, ..., 10^{-5}\}$ to identify the most effective value for each algorithm, using Eq. 5, 6 and 7 as indicators for unlearning performance. All unlearning algorithms begin from the same original model for every dataset. We maintain the default unlearning hyper-parameters, i.e., the KL temperature to one, the confusion fraction for SFTC to zero and the k parameter for both CF-k and EU-k to five. For each algorithm, we establish a range of 1 to 4 epochs for the unlearning process to ensure that no algorithm exceeds 15% of the time required for complete retraining using the Adam optimizer and a batch size of 64. We keep the unlearned models at the epoch that achieved the best overall results across the five different trials. Finally, we report the average scores obtained from the most effective setting for each algorithm.

Table 1 presents the comparative analysis for each unlearning algorithm across the three considered datasets and two model architectures. The highest performing unlearning algorithm for each metric is denoted with bold and the second best with an underline. The original model's (ORI) metrics are included as a reference.

The evaluation of unlearning algorithms' efficacy requires considering all metrics as a whole, i.e., we expect a low accuracy error, high JS divergence and high MIA efficacy. For instance, if the original model remains unchanged, it exhibits no accuracy loss.



(a) Convergence on CIFAR-10. The target accuracy for the test and forget sets are 0.89 and 0.9, respectively.

(b) Convergence on MUFAC. The target accuracy for the test and forget sets are 0.59 and 0.49, respectively.

(c) Convergence on FER. The target accuracy for the test and forget sets are 0.63 and 0.21, respectively.

Figure 4: Unlearning Algorithms Convergence. The blue line corresponds to the retain set, the green line to the test set and the red line to the forget set.

However, this would also lead to no distinguishability as well as poor unlearning certifiability with respect to MIA.

SFTC emerges as the most effective unlearning algorithm, having the highest quality in 12 individual cases and the second best in 14. BD is the next most successful, performing the best in 7 individual cases and second best in 9 cases. The similarity in performance between SFTC and BD is expected since they optimize the same KL term. However, SFTC's integrated mechanism of targeted confusion and selective fine-tuning further enhances unlearning effectiveness.

On the ResNet-18 architecture, SFTC consistently ranks as the top or second best algorithm across all datasets, indicating robust unlearning. BD serves as the second most effective. Other algorithms like SCRUB and EU-5, show effectiveness in specific cases, such as low accuracy loss (MUFAC-SCRUB) and high distinguishability (FER-EU-5). Nevertheless, SFTC and BD emerge as the top-performing algorithms when all metrics are considered collectively.

For the EfficientNet model, while SFTC and BD maintain their high quality unlearning performance, the results show variability across datasets. For instance, in FER and MUFAC, the FT baseline leads to high quality, having the least accuracy loss with respect to the retrain-from-scratch oracle and high MIA efficacy in FER. Yet, SFTC and BD remain the most consistently effective algorithms.

Most algorithms demonstrate high utility in terms of MIA, substantially surpassing the original model, which fails to offer any unlearning certifiability. In many cases, they also exceed the utility of the retrain-from-scratch oracle, suggesting that unlearning algorithms can also mitigate issues like overfitting and model biases.

In Fig. 4, we present the convergence of the considered unlearning algorithms using a consistent trial with the same random initialization on the ResNet-18 model. For this experiment, we terminate the unlearning algorithm when the accuracy for $\mathcal{D}_t$ and $\mathcal{D}_f$ closely approaches the corresponding retrain-from-scratch accuracies.

For the FT baseline across all datasets, an initial decrease in accuracy in the first epoch is evident, followed by subsequent tuning towards $\mathcal{D}_r$. Similar patterns are observed with the CF-5 and EU-5 approaches, where the initial epoch noticeably diverges the model from its original state. The NG+ algorithm demonstrates a consistent trend across all datasets, seemingly leading to a global forgetting, which is demonstrated by a reduction in accuracy for all sets. This suggests that NG+ induces global forgetting, rather than being limited to the $\mathcal{D}_f$ alone.

SCRUB, in the CIFAR dataset, mirrors the NG+ pattern, reducing accuracy across all sets. In MUFAC, SCRUB initiates with a drop in all sets, subsequently elevating accuracy on $\mathcal{D}_r$, with test accuracy remaining consistent. On the other hand, accuracy on $\mathcal{D}_f$ displays variability across epochs initiating with a drop, elevating closely to $\mathcal{D}_r$ and then dropping near the retain-from-scratch target accuracy. In FER, SCRUB lowers $\mathcal{D}_r$ accuracy across epochs while the accuracy for $\mathcal{D}_t$ presents stability. Meanwhile, accuracy on $\mathcal{D}_f$ demonstrates a consistent decline, with an uptick noted at epoch 4.

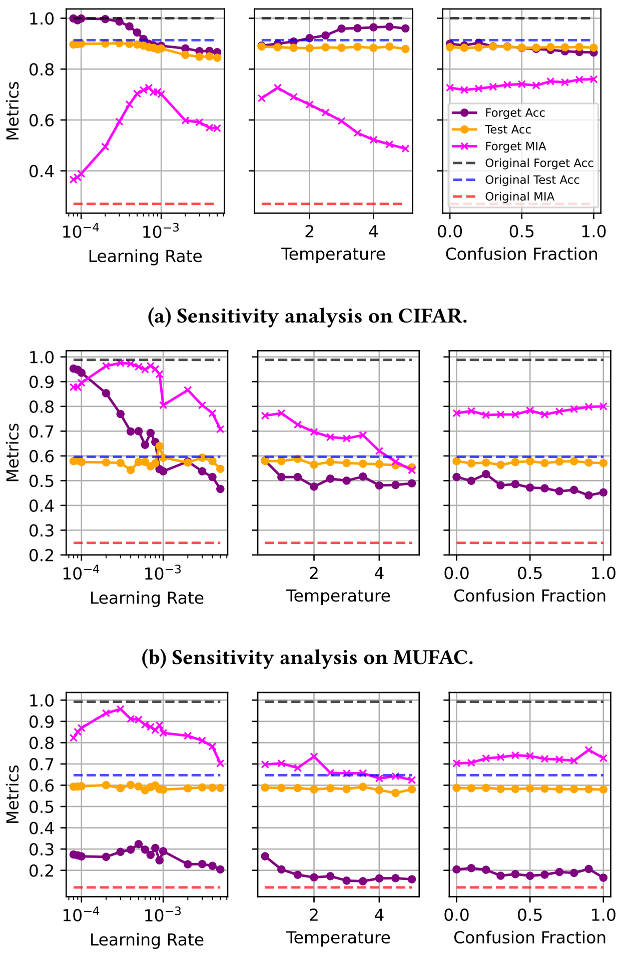
For the BD and SFTC algorithms, we observe a similar pattern in CIFAR, with both models lowering the $\mathcal{D}_f$ accuracy by approximately 10% and $\mathcal{D}_t$ accuracy by 1%, aligning with the target model's respective accuracies. In MUFAC and FER, BD lowers the predictive performance across all sets, with a higher impact on $\mathcal{D}_t$ compared to the retrain-from-scratch model. In contrast, SFTC begins by reducing accuracy in MUFAC below the target, but by epoch 4, it surpasses BD in terms of target accuracy resemblance. In FER, SFTC's performance is akin to BD, albeit with closer $\mathcal{D}_t$ and $\mathcal{D}_f$ accuracies to the target. These observations suggest that following a completely random model for the forget set (BD) negatively impacts a broader sample range. Hence, adopting a biased random model towards the correct labels for the forget set (SFTC) facilitates more precise forgetting in alignment with the target model. The effectiveness of the biased random model approach will be further clarified in the subsequent sensitivity analysis study.

5.6 Sensitivity Analysis

In this section, we conduct a sensitivity analysis to assess the impact of three hyper-parameters on the training dynamics of SFTC, i.e., learning rate, KL divergence temperature (τ) and confusion fraction (c). Fig. 5 presents the results regarding unlearning accuracy on $\mathcal{D}_f$ and $\mathcal{D}_t$ as well as the MIA efficacy for $\mathcal{D}_f$ (as defined in Eq. 9). We employ the ResNet-18 model, setting the number of epochs to two for CIFAR (Fig. 5a) and FER (Fig. 5c) and three for MUFAC (Fig. 5b).

Learning Rate. We begin by applying different learning rates in the range $[8 \times 10^{-5}, 9 \times 10^{-5}, \ldots, 5 \times 10^{-3}]$, fixing τ and c to 1 and 0, respectively (i.e., no temperature and biased output towards the correct label for $\mathcal{D}_f$). For CIFAR, lower learning rates (8×10^{-4} to 5×10^{-4}) are insufficient to induce forgetting since they marginally reduce the accuracy on $\mathcal{D}_f$. This pattern is also presented in MIA terms, where low MIA scores indicate that samples from $\mathcal{D}_f$ are predicted as members. Conversely, learning rates between $6 \times 10{-}4$ to 10^{-3} result in a desirable balance, lowering the accuracy on $\mathcal{D}_f$ and maintaining high accuracy on $\mathcal{D}_t$. Similarly for MIA, there is an upward trend, indicating higher unlearning effectiveness. Learning rates above $2 \times 10{-}3$ cause a drop in both $\mathcal{D}_f$ and $\mathcal{D}_t$ accuracies, indicating global forgetting, not tailored towards the forget set.

In MUFAC, similar to CIFAR, there is a decline in the forget set accuracy as the learning rate increases. Nevertheless, there is no clear optimal learning rate range when considering the balance between accuracy and MIA efficacy. Learning rates between 2×10^{-4} to 8×10^{-4} achieve high MIA (>95%) but lower accuracy on $\mathcal{D}_t$ compared to the retrained-from-scratch oracle. This indicates a trade-off between the (unknown) target and unlearning accuracy. An optimal setting for MUFAC, considering a real-world scenario, where the retrain-from-scratch oracle is unavailable, lies around 9×10^{-4}, achieving balance in $\mathcal{D}_t$ and $\mathcal{D}_f$ accuracies (0.5465 and 0.6393, respectively) as well as high MIA (0.93). However, this setting results in a high similarity error when compared to the retrain-from-scratch oracle. On the other hand, considering the optimal model as the most relevant with respect to the retrain-from-scratch accuracy (4×10^{-3}), this comes at the expense of MIA. Thus, a crucial open question is whether a retrain-from-scratch model should be used as reference across diverse datasets for evaluating unlearning.



(a) Sensitivity analysis on CIFAR.

(b) Sensitivity analysis on MUFAC.

(c) Sensitivity analysis on FER.

Figure 5: Sensitivity Analysis Results.

For FER, similar to MUFAC, lower learning rates result in higher MIA efficacy. We attribute this behavior to the dataset imbalance nature in contrast to the balanced CIFAR. The accuracy on $\mathcal{D}_t$ remains stable across learning rates, demonstrating SFTC's robustness. This is similar for $\mathcal{D}_f$, where unlearning is induced across the ranges of learning rates. The closest alignment with the retrain-from-scratch oracle is achieved with higher learning rates (e.g., 5×10^{-3}), at the cost of reduced MIA efficacy, similar to MUFAC. These findings suggest the need for further investigation into machine unlearning evaluation criteria without relying on a retrain-from-scratch oracle.

KL Temperature. To assess the impact of KL temperature, we conduct experiments using $\tau \in [0.5, 5]$ with a 0.5 step and keep the learning rate for CIFAR to 7×10^{-4}, MUFAC to 4×10^{-3} and FER to 5×10^{-3}. These values were the most optimal with respect to Eq.

5, 6 and 7 regarding the model's accuracy and did not consider the MIA efficacy. In all datasets, the accuracy on $\mathcal{D}_t$ does not present high variations and remains constant across different temperature values. In CIFAR, higher τ values lead to increased accuracy on $\mathcal{D}_f$, while in MUFAC and FER, a reverse trend is evident. Interestingly, a temperature of 1 consistently results in high MIA efficacy, suggesting that SFTC's default $\tau = 1$ is robust and effective for inducing unlearning, without needing precise temperature adjustments.

Confusion Fraction. Recall that under SFTC, the model tries to follow a biased output generation when considering $c < 1$ and a completely random output when $c = 1$ (similar to BD [4]). Our intuition was that by following a completely random output as in BD or by maximizing a loss term as in SCRUB, the model can be affected in a larger fraction of samples, not only those in $\mathcal{D}_f$, leading to decreased unlearning performance. Across all datasets, as c increases, the accuracy on $\mathcal{D}_f$ decreases, while the corresponding $\mathcal{D}_t$ remains stable with a slight decline. This highlights the potential benefits of using a biased output generator. Another interesting observation is that as MIA increases, the forget set accuracy decreases, indicating a trade-off between accuracy and MIA efficacy. This behavior is expected since the model loses more information regarding $\mathcal{D}_f$ as accuracy decreases, thereby increasing MIA efficacy. However, the optimal unlearned model lies between these two aspects, suggesting that incorporating such information during training could lead to improved unlearning algorithms.

6 CONCLUSION

In this work, we present a novel algorithm that refines an original model by fine-tuning it on the retain set while selectively confusing it through a biased random generator on the forget set. Our approach is evaluated on three diverse datasets using two deep neural network architectures for image classification tasks. Our results demonstrate that SFTC effectively induces forgetting and serves as one of the most promising unlearning algorithms compared to similar methods in terms of unlearning effectiveness, certifiability and verification. In addition, we present a realistic forget set for the FER-2013 dataset, tailored to include contextual information. Our findings highlight the variability in the performance of unlearning algorithms across different dataset types (balanced vs imbalanced) and illustrate a trade-off between the effectiveness of unlearning in maintaining both high accuracy and privacy preservation.

In the future, we aim to explore the effectiveness of SFTC on additional datasets including *tabular*, *language* and *graph-based* data. To demonstrate the generalization and scalability of machine unlearning it is crucial to encompass diverse tasks, such as *regression* and *recommendation*. Another critical aspect is the definition of novel *unlearning metrics* that do not rely on a retain-from-scratch oracle, which in real-world scenarios cannot be obtained. Lastly, another dimension is to evaluate machine unlearning under *differentially-private models* to provide insights on the dynamics of unlearning algorithms within environments that prioritize high privacy levels.

ACKNOWLEDGMENTS

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

REFERENCES

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on SP*. IEEE, 141–159.

[2] Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on SP*. IEEE, 463–480.

[3] Dasol Choi and Dongbin Na. 2023. Towards Machine Unlearning Benchmarks: Forgetting the Personal Identities in Facial Recognition Systems. *arXiv preprint arXiv:2311.02240* (2023).

[4] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. 2023. Can bad teaching induce forgetting? Unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7210–7217.

[5] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, New York, NY, USA, 1322–1333.

[6] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making AI Forget You: Data Deletion in Machine Learning. In *Advances in Neural Information Processing Systems*, Vol. 32.

[7] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. 2022. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640* (2022).

[8] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9301–9309.

[9] Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, et al. 2013. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013.* Springer, 117–124.

[10] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference*, Vol. 35. 11516–11524.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

[12] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. 2023. Machine Learning Operations (MLOps): Overview, Definition, and Architecture. *IEEE Access* 11 (2023), 31866–31879.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[14] Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. 2023. Towards Unbounded Machine Unlearning. In *NeurIPS 2023*. PMLR.

[15] Ananth Mahadevan and Michael Mathioudakis. 2021. Certifiable machine unlearning for linear models. *arXiv preprint arXiv:2106.15093* (2021).

[16] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299* (2022).

[17] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2022. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Comput. Surv.* 55, 6, Article 114 (dec 2022), 29 pages.

[18] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems* 31 (2018).

[19] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on SP*. 3–18.

[20] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[21] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*. 4007–4022.

[22] Enayat Ullah, Tung Mai, Anup Rao, Ryan A. Rossi, and Raman Arora. 2021. Machine Unlearning via Algorithmic Stability. In *Proceedings of Thirty Fourth Conference on Learning Theory*. PMLR, 4126–4142.

[23] Eduard Fosch Villaronga, Peter Kieseberg, and Tiffany Li. 2018. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review* 34, 2 (2018), 304–313.

[24] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. 2023. Machine Unlearning: A Survey. *ACM Comput. Surv.* 56, 1, Article 9 (aug 2023).

[25] Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. 2023. A review on machine unlearning. *SN Computer Science* 4, 4 (2023), 337.