Empirical Analysis of Machine Unlearning Methods

Akshat Naik, Leen Lababidi, Michael Rubenstein

1 Introduction

Deep learning has led to some of the most impressive technological advancements today, having a wide array of applications from realistic image generation, to a dialogue system that can hold human-like conversations. A central factor in this success has been the availability of large amounts of data, computing power, and the ability of deep learning methods to learn intricate patterns from the data.

Much of this data also contains personal data that raises privacy concerns. For instance, research [1, 2] has shown that in some cases, using membership inference attacks (MIAs), one can detect with high accuracy whether an example was used to train a machine learning model. This implies that even if an individual's data is deleted from a database, it may still be possible to infer whether the individual's data was used to train the model or not.

Moreover, an increasing number of privacy laws are targeting machine learning applications. For example, Article 17 of the General Protection Regulation (GDPR) of the European Union, *the right to be forgotten*, requires machine learning applications to remove a portion of the data from a dataset and retrain it if a user makes such a request.

Machine unlearning is a new paradigm in the field of machine learning that aims to remove the influence of a specific subset of training examples—the **forget set**—from a trained model, while ideally still maintaining other beneficial properties of the model such as the accuracy on the rest of training set and generalization over unseen examples.

A simple solution may be to simply train the model from scratch with the training data that excludes the desired forget set. However, this can be computationally expensive. An ideal unlearning algorithm would use the pre-trained model as a starting point and make adjustments to remove the influence of the forget set efficiently.

1.1 Problem Definition

In August of 2023, Google held a research competition with the goal of tackling the machine unlearning problem [3] [4]. In this project, our goal is to empirically evaluate how well the top three algorithms from the competition perform when some of the conditions set during the competition are relaxed. Specifically, the models submitted by Kaggle users Fenchuan [5], Doun Lee [6], and Seif Eddine Achour [7]. Each submission has a publicly-available link to a notebook containing their solution.

1. **Fenchuan's:** This solution consisted of two stages. The first, *Forget Stage*, ran for one epoch and minimized the KL-divergence between the output logits and a uniform pseudo label on the forget set. Secondly, it ran an *Adverserial Fine-tuning Stage*, through which it alternated between "forgetting" and "retaining." When forgetting, the model maximized the difference between the logits of the two sets, and when retaining, the model is simply fine-tuned on the retain dataset again [5].
2. **Doun Lee's:** First, *collect gradient information* from both sets, using gradient ascent and descent on the forget and retain sets respectively. Then, *initialize the weights* of the convolution filter based on these gradients. Finally, *retrain* the model on the retain set [6].
3. **Seif Eddine Achour's:** Train the model on the retain set normally, then insert some noise to the weights before the final epoch, with the goal of forgetting the precise details but retaining the "overall" global information. This solution also used a weighted cross-entropy loss function to account for the heavy class imbalance in the original data [7].

We will train and run each of the models described above, then evaluate its performance when the following are varied:

- **Forget Set Size:** During the competition proceedings, the forget set size was fixed to 2% of the training data. It's intuitive to assume that as the size of the forget set increases, the unlearning algorithms may perform poorly. We wish to identify any differences in performance between the three winners under three sizes of the forget set: 1%, 10%, 20% and 50%. Additionally, we will vary the composition of the forget set.
- **Forget Set Composition:** The forget set was composed mainly of data from the first two age classes. Some iterations that may be worth investigating are choosing an equal number of samples from each of the classes, choosing only from classes that were not included in the competition, and choosing a distribution demographically proportional to the training data.
- **Training Data Class Imbalance:** The vast majority of the training dataset used belonged to the first age class. We want to test how the forget quality changes if the class distribution was more balanced by running the algorithms on a truncated training set.

To ensure a standard basis of comparison, we will aim to use a similar evaluation as the competition. Unfortunately, some of the evaluation methods using in the competition are still kept secret, even after the conclusion of the competition so we couldn't replicate the exact evaluation methods. Moreover, some parts of the competition evaluation method required training from scratch but with different seeds over 500 times for each forget set, and with different forget sets this number could go up to thousands of times. So we chose instead to come up with a similar evaluation method (explained later) which is more computationally feasible for us.

The dataset they used for the competition was Face Synthetics dataset from Microsoft [8]. These are 100,000 images of synthetic faces. The competition coupled these faces, with their own annotations of each face being put into an age bin, or a range of ages (for eg. a bin of 23-29 ages). Unfortunately, these annotations were also secret and not released to the public. The participants had to submit their unlearning code that the Kaggle site ran under the hood with the data and their labels. So we were forced to choose another dataset for our needs. We ended up choosing UTKFace because that fit our criteria of being a large enough dataset (20,000 images), being annotated with age, as well as some other features like gender and race [9]. We used the Aligned & Cropped version of the dataset so we could focus on the face of the person and had the same image size for each image.

2 Prior Work

The rise in legislation concerning user privacy highlights the importance of addressing the problem of data deletion from already trained machine learning models.

Since retraining the model entirely is too expensive, the current focus of machine unlearning is to develop algorithms and methods that don't require retraining of the model. The current field of machine unlearning is split between two main objectives [10]:

- **Exact Unlearning:** Ensuring that the new model is exactly the same as one that was trained without data in the forget set.
- **Approximate Unlearning:** Making an approximation of the model that was trained without data in the forget set.

The majority of the field is trying to maximize the latter objective, we will also be evaluating models that also focus on the latter objective.

Mahadevan et al. [11] groups these approximate unlearning methods into three categories:

1. This method **utilizes the remaining data in the training dataset**. They use the remaining data and Fisher Information to add optimal noise to the models weights to ensure that the deleted datas influence is minimized.
2. This method **utilizes the forget set data**. They perform a Newton step to approximate the datas impact on the model and try to remove it. This method also adds noise to the model to ensure certifiably.
3. This method **utilizes information stored during model training**. It tries to approximate full retraining SGD steps on the new dataset by using intermediate quantities from previous training such as gradients and model updates from each SGD step.

The top three methods of the Unlearning competition roughly fall under the first two categories as they utilize both the retain set and the forget set data, however the way the use that data is different from what Mahadevan et al. suggested.

3 Methods

3.1 Training our ResNet18 Model from scratch

In this section, we elaborate on our methodology for training the ResNet18 model, encompassing key decisions regarding hyperparameters, optimizer selection, and training strategy.

The dataset used in the Google competition lacked detailed documentation. However, it was observed that the age labels were organized into bins of size 6 for age prediction, rather than individual ages. Following this approach, we segmented our UTKFace age labels into 20 bins of equal size.

Instead of employing a pretrained model and fine-tuning, we opted to train our model from scratch. This decision was driven by the belief that training from scratch would enable our model to better adapt to our specific task of age prediction from human faces. Unlike ImageNet-1K, which contains diverse objects such as "acorn" and "school bus," our task focuses solely on facial attributes. Training from scratch allows the model's convolution filters to extract features tailored to our task, potentially leading to improved performance.

Our model architecture remained consistent with the competition's methodology, employing ResNet18 as the underlying model and customizing it by replacing the final fully connected layer. This updated layer comprised 20 neurons, intended to predict logits for each age bin. Subsequently, these logits were passed through a softmax layer to obtain probabilities for each age bin.

During the training process, we optimized the model parameters using cross-entropy loss on the softmaxed predictions. The following hyperparameters were employed:

1. **Learning Rate:** 0.01
2. **Weight Decay:** 0.0001
3. **Batch Size:** 100
4. **Number of Epochs:** 30

We adopted the **Adam** optimizer, maintaining the same learning rate and weight decay as described above throughout training.

During the training iterations, we observed fluctuations in the loss function. To address this issue, we implemented a **learning rate scheduler**, reducing the learning rate by a factor of 0.1 every 5 epochs. This adjustment, coupled with the weight decay, contributed to stabilizing the learning process and facilitating model convergence.

3.2 Evaluation Methods

Taking our starting model which was trained on the entire training set, we obtain an unlearned model by applying one of the unlearning algorithms given the forget set, and the training set. We tested the unlearned model on that forget set by comparing it to a model that was trained only on the retain set (which is the training set excluding the forget set). This way, the retrained model acts as an oracle (or an ideal) to the unlearned model.

There were three parts to the score:

- Effectiveness (Test): This score is obtained by dividing the accuracy of the retrained model on the test set by the accuracy of unlearned model.
- Effectiveness (Retain): This score is obtained by dividing the accuracy of the retrained model on the retain set by the accuracy of the model after an unlearning algorithm has been applied.
- Certifiability (Forget): This score is obtained by taking the accuracy of the retrained model and the unlearned model, and dividing the minimum of those two by the maximum of them.

All these parts were partially inspired by what limited information we had about the evaluation metric in the competition, as well as the evaluation methods suggested in Haibo Zhang et al. [12].

The first two scores ensured that the unlearned model keeps its desirable properties like generalizing to the test set and still maintaining a decent accuracy on the retain set. The last score penalizes the unlearned model if there is too much disparity between the accuracies of the retrained model and of the unlearned model. The reasoning behind this is that if the accuracy of the unlearned model differs too much from the relearned model, then the influence of the forget set is still prevalent in the unlearned model.

We combined the three scores by multiplying effectiveness (test) and effectiveness (retain) by the square of certifiability (forget), to a get a combined score. We used squared the certifiability to emphasize how important it was to remove the influence of the forget set, and not let the score be dominated by the effectiveness (test) and effectiveness (retain) which could be reasonably maximized by having an unlearning algorithm do absolutely nothing, as the original model would be expected to have the highest accuracy on test and retain set.

3.3 Experiments with different forget set

In this section, we describe our experiments involving variations in the forget set, focusing on adjustments in size and proportions of gender and race probabilities. Our dataset included information on age, gender (male/female), and race (white/black/asian/indian/other).
To manipulate the forget set, we varied its size and the proportions of male and race probabilities individually, while keeping the other variables constant. The default settings for each variable were as follows:

- Default Size: 2% of the train dataset (consistent with the competition)
- Default Gender Proportions: 50/50 (equal split)
- Default Race Proportions: 20/20/20/20/20 (equal split)

The variations for each of the variables are the following:

- Size Variations: 1%, 2%, 5%, 10%, 20%, 50%
- Gender (Male/Female) Proportion Variations: 30/70, 50/50, 70/30
- Race (White/Black/Asian/Indian/Other) Proportion Variations:
  - 20/20/20/20/20
  - 60/10/10/10/10
  - 10/60/10/10/10
  - 10/10/60/10/10
  - 10/10/10/60/10
  - 10/10/10/10/60

This results in a total of 15 variations in the forget set configuration. For each variation, we maintained the default settings for the variables not under consideration. For example, when examining size variations, we kept the gender proportions and race proportions at their default values. Similarly, when exploring gender proportion variations, we held the forget set size and race proportions constant. This approach ensured that we could isolate the effects of each variable on the forget set's behavior while maintaining consistency in the other aspects of the experiment.

3.4 Files and Code

For the code aspect of the project, we provide three Colab files:

1. **DataCleaningModelTraining.ipynb**: This notebook contains functions related to data cleaning and model training. It includes code for preprocessing the data and training both the base model and models trained on the retain set.
2. **Project.ipynb**: This notebook encompasses the implementation of the top three competitors' unlearning algorithms, evaluation metrics, and a function for executing experiments. It serves as the main script for conducting experiments and evaluating the performance of different approaches.

3. **PlotResults.ipynb**: This notebook includes the implementation for plotting results of the accuracies obtained. It generates four types of graphs (per variation type, e.g., size, gender, race) which are bar graphs comparing Certifiability Score, Retain Effectiveness Score, Test Effectiveness Score, and the combined Accuracy Score, resulting in a total of 12 graphs.

These Colab files encapsulate the core functionalities of our project, facilitating data processing, model training, experimentation, and evaluation.

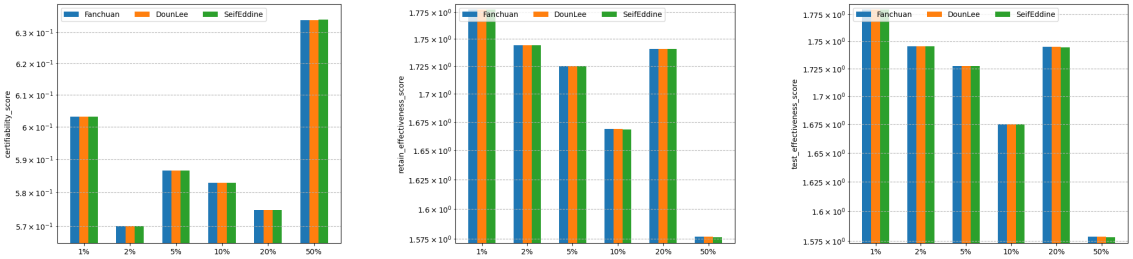
3.5 Running the Code

This section outlines the basic steps for running the code on Google Colab.

1. Download the UTKFace Dataset and unzip the file in your desired directory.
2. In each of the files, change the `ROOT_FOLDER` variable to the location of the directory that holds the unzipped dataset directory.
3. If you want to use Colab's GPUs, set the `IS_MAC` variable to `True`; if you are training locally, set `IS_MAC` to `False`.
4. Now, you can simply click "Run All" for the code to execute in each file without needing to change anything else!

4 Results

4.1 Score Graphs



(a) Certifiability Score **(b)** Retain Effectiveness Score **(c)** Test Effectiveness Score

Figure 1: Score Comparisons Across Size Ratios: 1%, 2%, 5%, 10% and 50% of Train

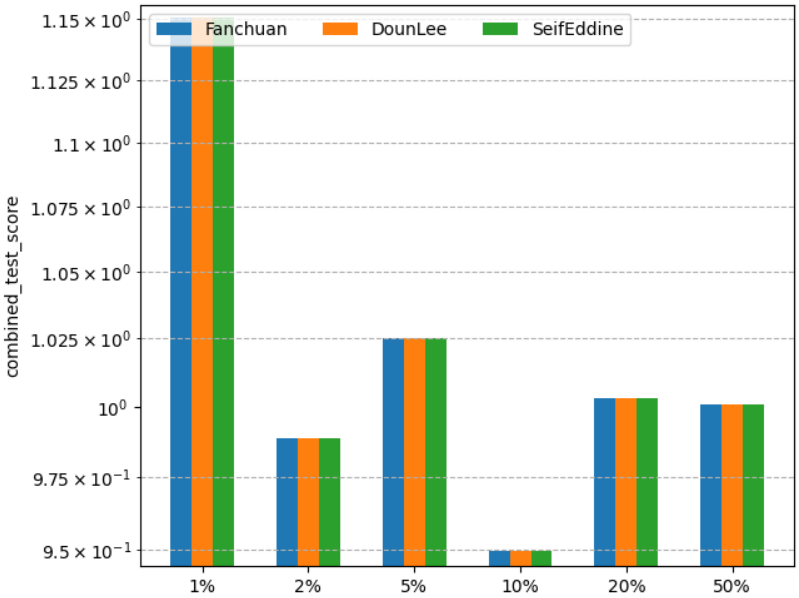
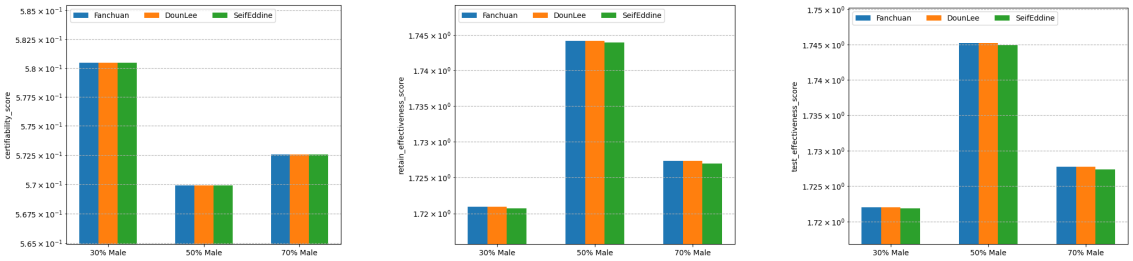


Figure 2: Combined Score Comparison Across Size Ratios: 1%, 2%, 5%, 10% and 50% of Train



(a) Certifiability Score **(b)** Retain Effectiveness Score **(c)** Test Effectiveness Score

Figure 3: Score Comparisons Across Gender Ratios: 30%, 50%, and 70% Male

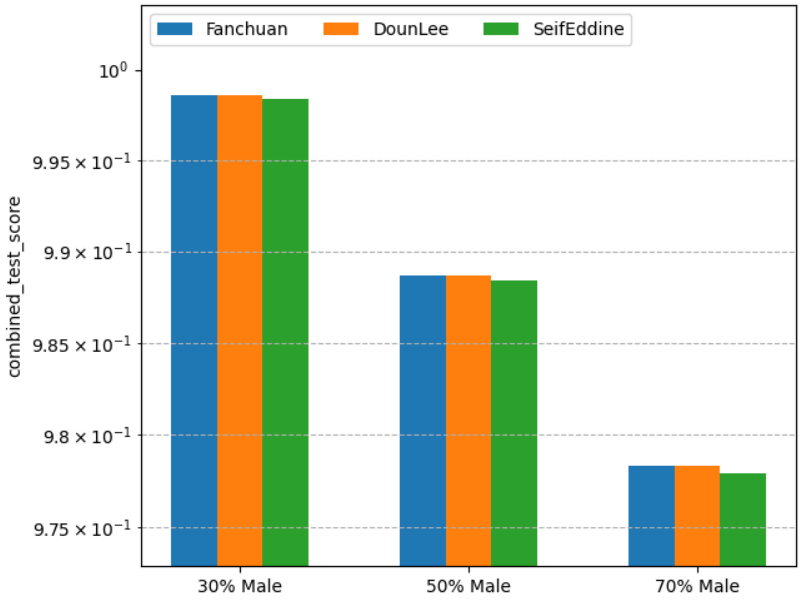
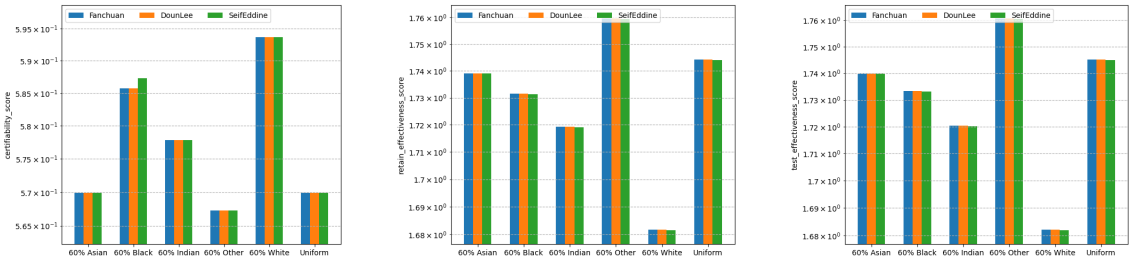


Figure 4: Combined Score Comparison Across Gender Ratios: 30%, 50%, and 70% Male



(a) Certifiability Score **(b)** Retain Effectiveness Score **(c)** Test Effectiveness Score

Figure 5: Score Comparisons Across Race Ratios

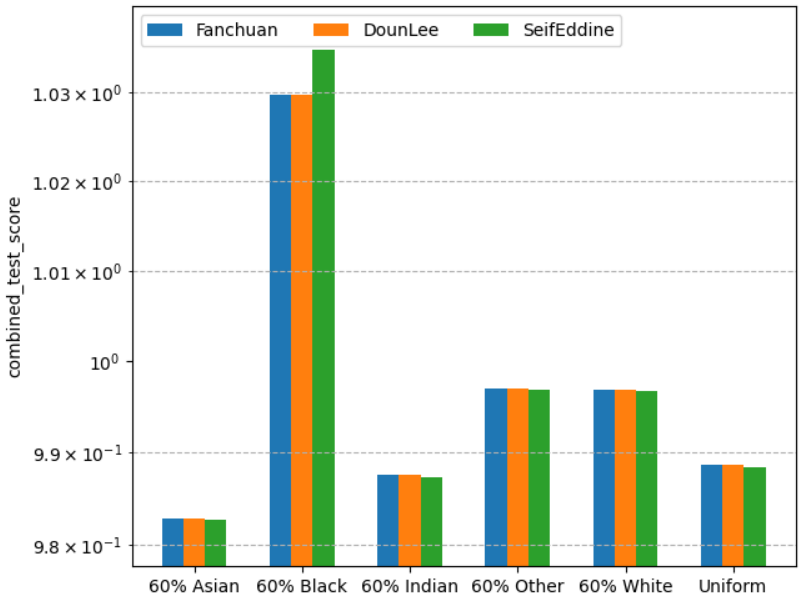


Figure 6: Combined Score Comparison Across Race Ratios

4.2 Discussion

Overall, the performance of all three methods remained extremely comparable across the various experimental variants. In the experiments involving different dataset sizes, there was minimal deviation in the scores obtained by each method. The Fanchuan method (#1) demonstrated slightly superior performance across different gender ratios while the SeifEddine method (#3) exhibited worse performance. However, when examining the race variant of experiments, the Fanchuan method consistently outperformed (albeit slightly) the other two methods across all racial compositions, except in scenarios where the dataset contained a majority of Black individuals, where the SeifEddine method emerged as the top performer.

Interpreting the exact output values for each experiment variant presents some challenges. Initial expectations assumed that the performance of the unlearning algorithms would either deteriorate or improve as the size of the forget set increased. However, observed results indicated varying performance trends across all three methods as the forget set size expanded. Furthermore, we don't have an easy explanation for the observation that all three unlearning algorithms performed better when the forget set comprised more females.

Given the comparable performance of all three algorithms across different demographic compositions, it is plausible to speculate that these results are more indicative of dataset characteristics rather than inherent features of the algorithms themselves. Despite significant differences in their underlying methodologies, the three algorithms exhibited similar performance trends, suggesting that dataset nuances might play a more substantial role in influencing outcomes than algorithmic variations.

5 Conclusion

All three methods performed very comparably to each other. The Fanchuan method slightly outperformed the two, but the performance advantage was small enough that more experiments on other datasets and domains will have to be conducted before we can declare a clear winner.

5.1 Limitations of our Experiments

- One notable limitation is that our dataset differs from the competition dataset. Some notable differences include:
  - Their dataset contained synthetic faces, while ours contained real human faces.
  - Their dataset contained 100,000 images, while ours contained only 20,000.

  The datasets' subject differences may result in the ResNet model not being expressive enough for our current task, as we only reached a training accuracy of around 88%, while the com-

petition reached a training accuracy of around 98%. Additionally, the smaller size of our dataset may limit the complexity and diversity of patterns learned by the model during training, potentially leading to suboptimal or different results.

- Another notable limitation is that some of the images in our UTKFace dataset were black and white, and some were extremely blurry. This may affect our model's ability to capture features, hence it may also affect the model's ability to forget said features when running the algorithms.
- The competition organizers didn't provide enough documentation on the evaluation metrics that they used for the competition, which they said they would after the competition ended. Hence, we could not make use some of the membership inference attacks that they used.
- Our results are not as robust as they could be because we could not rerun the experiments on different random seeds due to time and computation constraints. Evaluation requires us to retrain a model from scratch on only the retain set, which takes time and resources.

5.2 Future Work

- Broaden the evaluation metrics beyond accuracy to provide a more comprehensive assessment of model performance. While accuracy is a commonly used metric, accessing additional evaluation metrics, including but not limited to Google's evaluation criteria, could unveil insights into various aspects of model behavior. These insights are crucial for understanding algorithm performance across different submissions from the top three competitors.
- Extend the application of unlearning algorithms to diverse domains such as language modeling to enhance the generalizability of our findings. By evaluating the effectiveness of unlearning techniques in various contexts, including language modeling tasks, we can draw conclusions on the most effective algorithms regardless of the specific domain. This broader investigation will yield valuable insights into the versatility and robustness of unlearning techniques.
- Investigate the adaptability of unlearning algorithms to different neural network architectures beyond ResNet18, encompassing both convolutional neural networks (CNNs) and non-CNN networks. Through experimentation with a diverse range of architectures, we can evaluate the performance of unlearning techniques across various model designs and uncover potential architectural dependencies. This exploration will deepen our understanding of how the effectiveness of unlearning techniques varies with architectural choices, thereby informing recommendations for their application in diverse modeling scenarios and potentially guiding the development of improved unlearning algorithms.

References

[1] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2017.

[2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles, 2022.

[3] Fabian Pedregosa and Eleni Triantafillou. Announcing the first machine unlearning challenge. `http://blog.research.google/2023/06/announcing-first-machine-unlearning.html`, 2023. Accessed: 26-02-2024.

[4] Jamie Hayes Peter Kairouz Isabelle Guyon Meghdad Kurmanji Gintare Karolina Dziugaite Peter Triantafillou Kairan Zhao Lisheng Sun Hosoya Julio C. S. Jacques Junior Vincent Dumoulin Ioannis Mitliagkas Sergio Escalera Jun Wan Sohier Dane Maggie Demkin Walter Reade Eleni Triantafillou, Fabian Pedregosa. Neurips 2023 - machine unlearning. `https://kaggle.com/competitions/neurips-2023-machine-unlearning`, 2023.

[5] Fanchuan. Neurips 2023 - machine unlearning competition first place winner submission. `https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458721`, 2023.

[6] Doun Lee. Neurips 2023 - machine unlearning competition second place winner submission. `https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459200`, 2023.

[7] Seif Eddine Achour. Neurips 2023 - machine unlearning competition third place winner submission. `https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/459334`, 2023.

[8] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021.

[9] Song Yang Zhang, Zhifei and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[10] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. `https://doi.org/10.48550/arXiv.2110.11891`, 2022. Accessed: 25-02-2024.

[11] Ananth Mahadevan and Michael Mathioudakis. Certifiable machine unlearning for linear models. `https://doi.org/10.48550/arXiv.2106.15093`, 2021. Accessed: 25-02-2024.

[12] Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine unlearning, 2023.