POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Bad Teaching in Machine Unlearning with Similarity-based Sampling

Supervisors

Dr. Flavio GIOBERGIA

Prof.ssa Elena Maria BARALIS

Candidate

Claudio SAVELLI

July 2024

Summary

The world of Artificial Intelligence (AI) and Machine Learning (ML) is constantly changing, and the concept of 'Machine Unlearning' has emerged as a challenging area of research. This concept is becoming increasingly relevant as the huge adoption of AI and ML technologies has introduced numerous ethical, moral, and privacy concerns, particularly regarding using personal data in training these models.

The central objective of Machine Unlearning is to erase the influence of specific data inputs from a model's training set. While it is relatively straightforward to do so from databases, erasing it from an AI model presents a significant challenge due to these models' complex nature. Furthermore, making a model forget certain information is particularly tough due to the stochastic nature characteristic of all Deep Neural Networks, including widely used models such as Large Language Models (LLMs). The stochasticity in the learning processes complicates the evaluation of the effects of specific data on the model's training phase. Moreover, the efficacy of Machine Unlearning techniques is difficult to evaluate. This difficulty is also raised by the lack of established metrics for evaluating these methods since, as already said, it is difficult to assess how much a model is influenced in its prediction by specific data, making the development of such metrics an open field of research.

This thesis aims to explore the domain of Machine Unlearning. It seeks to analyze the underlying dynamics of various unlearning approaches, examining some strategies employed to remove influences of specific data from models and the principal metrics used to assess their effectiveness. In addition to the existing methodologies, this research includes the development of a new unlearning method proposed by the author, which has been shown to outperform the other described techniques considering the analyzed metrics. A new benchmark has been proposed, which includes the definition of three new datasets. The goal is to create a shared framework for effectively comparing different unlearning techniques. By doing so, the research intends to contribute to active and emergent discussions in the academic community, fostering a better understanding of how unlearning impacts model integrity and data privacy. This exploration is particularly pertinent as the demand for ethical AI solutions becomes crucial for everyday applications.

Table of Contents

1 Introduction 1
1.1 Machine Unlearning . . . 1
1.1.1 Reasons for Machine Unlearning . . . 1
1.1.2 Formal Formulation of Machine Unlearning . . . 2
1.1.3 Approaches to Machine Unlearning . . . 3
1.1.4 Challenges in Machine Unlearning . . . 5
1.2 Contributions of the Work . . . 6

2 Related Works 8
2.1 Methods . . . 8
2.1.1 Competent and Incompetent Teachers Method . . . 8
2.1.2 Scrub (SCalable Remembering and Unlearning unBound) . . . 9
2.1.3 Fine-tuning . . . 10
2.1.4 NegGrad (Negative Gradient Ascent) . . . 10
2.1.5 Advanced NegGrad with Classification Loss . . . 11
2.1.6 CF-k (Class-wise Forgetting) . . . 11
2.1.7 UNSIR (Unlearning by Selective Impair and Repair) . . . 11
2.2 Metrics . . . 12
2.2.1 Model's Utility . . . 12
2.2.2 Method's Efficiency . . . 12
2.2.3 Membership Inference Attacks (MIA) . . . 13
2.2.4 Forgetting Score . . . 13

3 Method proposed 14
3.1 Data Selection . . . 14
3.1.1 Similarity-Based Sampling (SBS) . . . 14
3.2 Unlearning Process . . . 16
3.2.1 Smart Batch Construction (SBC) . . . 17

4 Datasets 19
4.1 Dataset Division in MU Framework . . . 19
4.2 Datasets Proposed . . . 20
4.2.1 Forget Set construction . . . 21
4.2.2 Models Training . . . 21
4.2.3 MUCelebA . . . 22
4.2.4 Modified MUFAC (MMUFAC) . . . 23
4.2.5 MUCifar-100 . . . 25

5 Experimental Results 27
5.1 Experimental Setup . . . 27
5.1.1 Unlearning Models . . . 27
5.1.2 Dataset Description . . . 27
5.1.3 Metrics for Evaluation . . . 28
5.2 Comparative Analysis . . . 28
5.2.1 Results on MUCelebA . . . 29
5.2.2 Results on MMUFAC . . . 29
5.2.3 Results on Modified CIFAR-100 . . . 29

6 Conclusions 33
6.1 Summary of Findings . . . 33
6.2 Limitations . . . 34
6.3 Future Work . . . 35

List of Tables 36

List of Figures 37

A Similarity-Based Sampling examples 39

Bibliography 41

Chapter 1

Introduction

The introduction to this work will describe the concept of Machine Unlearning (MU), why it is used, and the current challenges. In addition, the main contributions made will be described.

1.1 Machine Unlearning

1.1.1 Reasons for Machine Unlearning

There are various fields and reasons in which Machine Unlearning is used. A short list of the main areas is shown below:

1. **Complying with the new AI regulations:** for worldwide regulations such as the GDPR (General Data Protection Regulation), Consumer Privacy Act, or Bill C-27, companies must ensure that their users have the 'Right to be Forgotten' [1], which is not only delete user data on demand from databases but also from Deep Learning models. The only solution for companies that want to follow these guidelines would be to retrain these models from scratch, removing the data to be deleted from the training. This method is impractical considering their high economic costs and environmental impact [2].

2. **Recovery of attacked models:** Machine Unlearning is crucial to recover machine learning models that have compromised poisoned training data points. Data poisoning attacks involve deliberately introducing malicious data into the training set, intending to corrupt the model's learning process and degrade its performance [3]. By effectively identifying these poisoned data points, Machine Unlearning techniques can mitigate the impact of such attacks, restoring the integrity and reliability of the affected models. By removing the influence of compromised data selectively, we can rehabilitate the model without retraining

it completely. This allows us to save computational resources and maintain the overall efficiency of the machine learning pipeline.

3. **Copyright claim:** Generative AI tools are trained on collections of material collected, usually by scraping the net. Some AI image and text generation tools have been trained on material taken from web pages without the consent or knowledge of the owners of those pages. It has not yet been openly stated whether using content by artists or writers without permission to train generative AI is copyright infringement. Should this be the case, however, Machine Unlearning may be an excellent solution to remove copyrighted data from model training without having to retrain all of it from scratch. For these reasons, Machine Unlearning on generative models of both text [4] and images [5] is another important field of research emerging in recent years.

4. **Ethical reasons:** Machine Unlearning could lead us towards developing models that ensure ethical and fairness principles, as it could help address biases and discrimination in AI systems by unlearning biased patterns and ensuring fair decision-making. It is well known that many datasets used to train models, including LLMs, contain heavily biased information. Once it is recognized which data leads to certain biases, thanks to unlearning approaches, it will be possible to remove those negative features from the model's training, making it fairer.

5. **Other applications:** Numerous other applications of Unlearning exist. However, for the sake of brevity, they are not exhaustively listed here.

1.1.2 Formal Formulation of Machine Unlearning

This section explores the basic principles of Machine Unlearning, which form the foundation for the practical applications discussed later. A figurative example of the Unlearning framework can be seen in Figure 1.1.

Let $\mathcal{D}$ represent the initial training dataset consisting of n data points, where each data point is denoted as $x_i \in \mathcal{D}$ for $i = 1, 2, \ldots, n$. A machine learning model $\mathcal{M}$ is trained on $\mathcal{D}$ to produce a set of learned parameters θ. The model $\mathcal{M}$ can be described as a function $f(\theta, x)$ that predicts outcomes based on input data x.

Given a subset $\mathcal{D}_f \subset \mathcal{D}$ of data points that need to be unlearned, usually denoted as *'Forget Dataset'*, the objective of Machine Unlearning is to update the model $\mathcal{M}$ such that its performance and internal state are as if $\mathcal{D}_f$ had never been part of the training data. Formally, let $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ be the remaining dataset after removing $\mathcal{D}_f$, usually denoted as *'Retain Dataset'*. The goal is to obtain a new set of parameters θ' such that the updated model $\mathcal{M}'$ with parameters θ' approximates a model trained exclusively with $\mathcal{D}_r$.

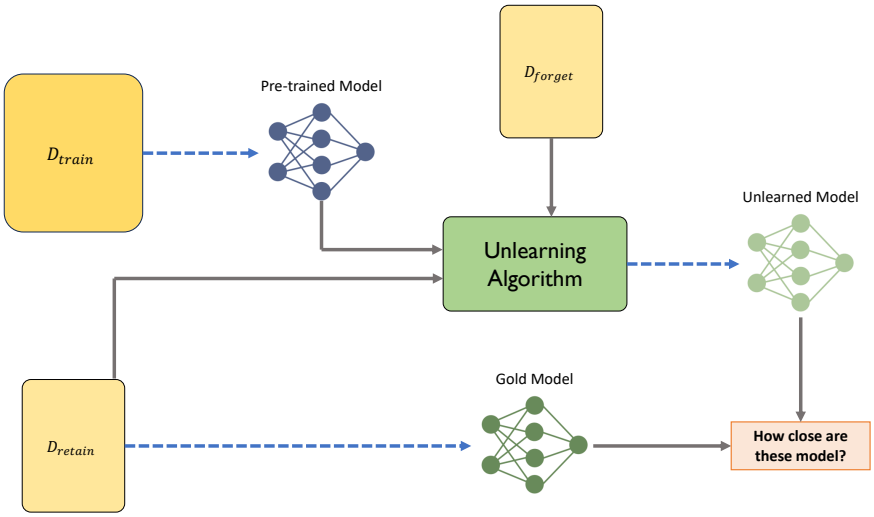

Figure 1.1: Anatomy of the Machine Unlearning framework

Exact unlearning, as will be seen in 1.1.3, aims to remove completely the influence of $\mathcal{D}_f$ from $\mathcal{M}$. This can be achieved through retraining or some algorithmic techniques. In the most straightforward approach, retraining, the model is retrained from scratch using only $\mathcal{D}_r$:

$$\theta' = \arg\min_{\theta} \sum_{x_i \in \mathcal{D}_r} L(f(\theta, x_i), y_i)$$

where L is the loss function used during training, and y_i are the corresponding labels. However, retraining is computationally and timely expensive [6].

Inexact Unlearning relaxes the requirement of exact equivalence between $\mathcal{M}'$ and a model trained solely on $\mathcal{D}_r$ by trying to update θ without full retraining. It seeks to minimize the influence of $\mathcal{D}_f$ while maintaining model performance. Some examples of Inexact Unlearning methods are listed in 2.1. Inexact Unlearning usually tries to dynamically update the model so that the system can adjust the model parameters θ to remove the influence of $\mathcal{D}_f$ without full retraining.

1.1.3 Approaches to Machine Unlearning

We will mainly analyze unlearning processes applied to Deep Neural Networks (DNNs), motivated by two main factors. Firstly, given their extensive use, DNNs are predominantly the object of study for unlearning techniques within the scholarly literature. Secondly, the relative ease of retraining shallow models reduces the need for unlearning, as these models require fewer resources to be trained from scratch.

As already anticipated in 1.1.2, unlearning techniques for DNNs generally fall into two algorithmic categories: exact and inexact. Exact unlearning guarantees

the complete removal of the influence of training data from the model, even though often at a high computational and financial cost. In contrast, inexact unlearning methods mitigate these costs but do not provide a formal guarantee of completely removing the influence of the data from the model. For this reason, it is crucial to develop metrics to assess how much specific training data influences a model's behavior to determine the goodness of the inexact unlearning method used.

Exact Unlearning Exact unlearning, also referred to as perfect unlearning, is defined as the process by which a machine learning model is modified to eliminate the influence of a specified subset of data, denoted as $\mathcal{D}_f$, from its learned parameters θ. The objective is to adjust the model so that its state and performance are indistinguishable from a model initially trained without $\mathcal{D}_f$. The goal of exact unlearning is to find a new set of parameters θ' such that the updated model $\mathcal{M}'$ with parameters θ' is equivalent to a model trained solely on $\mathcal{D}_r$.

The challenge of exact unlearning lies in its requirement for the new model $\mathcal{M}'$ to behave as if it had never seen $\mathcal{D}_f$. However, retraining the model from scratch using only $\mathcal{D}_r$ is often computationally expensive, especially for large-scale models and datasets. Therefore, more efficient algorithmic approaches have been developed to approximate the effect of retraining without incurring the total computational cost.

SISA (Sharded, Isolated, Sliced, and Aggregated) [7] is an example of an exact unlearning technique. This method partitions the training data into several disjoint shards, each of which independently trains a separate model. This sharding strategy enables unlearning by retraining only the slices affected by data removal rather than the entire model. The final model output is then obtained by aggregating the individual slices' predictions. An example of the SISA framework is shown in Fig. 1.2. SISA has many limitations anyway, such as potential reductions in model efficiency due to isolated data shards, increased computational and storage demands for managing multiple models, challenges in scaling, potential privacy vulnerabilities within shards, and a lack of flexibility in adapting to data distribution or model requirements.

The feasibility and efficiency of exact unlearning depend on the model's nature, the data's complexity, and the underlying machine learning algorithms.

Inexact Unlearning Inexact unlearning, also referred to as approximate unlearning, is defined as the process by which a machine learning model is modified to reduce the influence of a specified subset of data, denoted as $\mathcal{D}_f$, from its learned parameters θ. The objective is to adjust the model so that the influence of $\mathcal{D}_f$ is minimized, although not necessarily eliminated, ensuring that the model's state and performance are sufficiently close to a model that was trained initially only using $\mathcal{D}_r$.

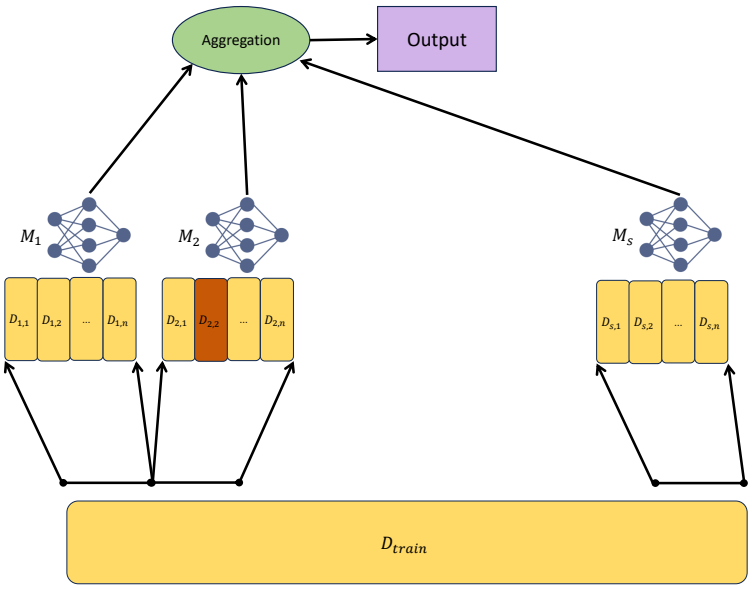

Figure 1.2: Anatomy of the SISA Framework [7]. The dataset D is partitioned into multiple shards $(D1, \ldots, Ds)$, each trained on separate models $(M1, \ldots, Ms)$. The red square highlights the specific data shard that needs to be unlearned. Outputs from all models are then aggregated to form the final model output

Formally, consider a machine learning model $\mathcal{M}$ with parameters θ, initially trained on a dataset $\mathcal{D}$. The goal of inexact unlearning is to find a new set of parameters θ' such that the updated model $\mathcal{M}'$ with parameters θ' closely approximates a model trained exclusively with $\mathcal{D}_r$ while acknowledging that some residual influence of $\mathcal{D}_f$ may remain.

Inexact unlearning relaxes the stringent requirement of exact equivalence and instead focuses on reducing the influence of $\mathcal{D}_f$ to an acceptable level. This relaxation allows for more computationally efficient approaches compared to exact unlearning. In inexact unlearning, a balance between computational efficiency and the degree of unlearning needs to be achieved. While it may not guarantee the complete removal of $\mathcal{D}_f$'s influence, inexact unlearning provides a practical solution for scenarios where exact unlearning is infeasible due to computational constraints.

1.1.4 Challenges in Machine Unlearning

Omitting the influence of specific data from DNN models is particularly difficult in the Machine Unlearning framework given their nature [6]. In addition, once the unlearning algorithm has been applied, assessing its goodness—measuring how much influence a specific data point still has within the model—remains one of the

most significant challenges in the field.

The main challenges of this framework are summarised as follows:

1. **Stochasticity of training:** Due to the stochastic nature of training, it is usually impossible to know when a data point was used to train a DNN model. In neural networks, the training dataset is partitioned stochastically into numerous subsets, each referred to as a 'batch'. This additional stochasticity makes assessing a data point's impact on the model's training even more complex.

2. **Incrementality of training:** The problem is enhanced by the incremental nature of the training process. Precisely, at a given time t, the data used to train the model influences the current state and all the subsequent updates. This dependence on the data used in t complicates the assessment of the impact that each data point may have on the complete model training. This dynamic introduces considerable complexity in understanding and predicting how much influence a data point has on a model.

3. **Catastrophic unlearning:** It is reasonable to expect that a model may lose accuracy after undergoing an unlearning process, as it loses information related to specific data points. Consequently, it is vital to evaluate the extent of accuracy loss when employing unlearning techniques, as this loss can lead to what is often indicated as 'catastrophic unlearning' [6]. Such significant degradation can make the model unusable. Therefore, monitoring and addressing this issue is critical in analyzing different unlearning frameworks.

4. **Evaluation of Unlearning:** As already described in 1.1.3, for the evaluation of the goodness of an inexact unlearning method, it is essential to have metrics that can assess the impact of a data point in the training of the model. Understanding the impact, as mentioned earlier, is not easy. In fact, currently, the metrics used in unlearning literature require a direct comparison of the unlearned model with the retrained one, making them not applicable in real-life scenarios. In addition, it is necessary to formalize these metrics to have a legally recognized guarantee [1] that the data to be forgotten have been sufficiently removed from the model [8].

   A more extensive description of some of the most currently used metrics can be found in 2.2.

1.2 Contributions of the Work

This work contributes significantly to the field of Machine Unlearning by addressing several key areas of research and development. The main contributions can be

summarized as follows:

- **Study of Machine Unlearning Frameworks:** An in-depth exploration and analysis of existing Machine Unlearning frameworks have been conducted. This study critically reviews current methodologies, metrics, and datasets and identifies areas where improvements are needed. This comprehensive analysis highlights the need for effective unlearning processes to comply with evolving norms and regulations.
- **Development of a Novel Unlearning Method:** A novel unlearning method is proposed, demonstrating higher performance than other unlearning techniques. This method is evaluated across different metrics, demonstrating its efficacy considering all the inexact unlearning requirements. Furthermore, thanks to the tunable parameters during the unlearning process, it is possible to find the right balance between data forgotten and the model's performance after unlearning, ensuring an optimal trade-off based on specific constraints and requirements.
- **Proposal of New Datasets:** Three new datasets—MUCelebA, Modified MUFAC, and MUCIFAR-100—are introduced to recognize the limitations of existing datasets in evaluating unlearning algorithms. These datasets are designed to create a general benchmarking environment that facilitates the comparison of unlearning methods across different scenarios. Each dataset tests different aspects of unlearning, including the effectiveness of privacy protection and the ability to handle complex data structures. More information about the considered datasets is present in Section 4.

Chapter 2

Related Works

From this point onward, the work will consider only inexact unlearning methods. This focus is due to their relevance and applicability in real-world scenarios where computational efficiency and scalability are crucial.

The following sections will discuss the most notable inexact unlearning techniques developed and recognized within the academic community for their effectiveness and innovation. Additionally, evaluating these methods relies on specific metrics, which are crucial for assessing how data has been effectively unlearned without significantly degrading the model's performance.

2.1 Methods

This section analyzes different inexact unlearning methods that have emerged as practical solutions for data removal in machine learning models. These methods are considered in the Results Section 5 to evaluate the novel unlearning method proposed.

2.1.1 Competent and Incompetent Teachers Method

Competent and Incompetent Teachers' unlearning method [9] uses a teacher-student framework. Here, knowledge is transferred to the student by both a competent teacher model ($\mathcal{T}_s$) and an incompetent teacher model ($\mathcal{T}_d$) as shown in Figure 2.1. The framework facilitates forgetting specific data by manipulating the student model's training process using accurate and random information.

Teacher-Student Configuration: Let the competent teacher $\mathcal{T}_c$ be a fully trained model on the full dataset $\mathcal{D}$, having parameters θ_c. The incompetent

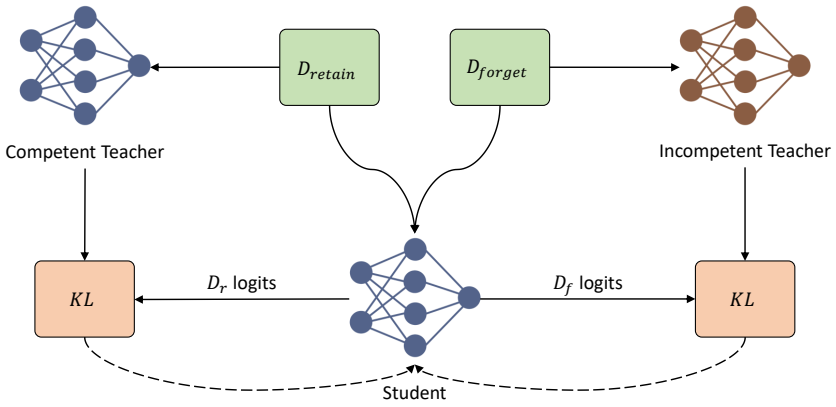

Figure 2.1: Anatomy of the Competent and Incompetent Teachers Framework [9]

teacher $\mathcal{T}_i$, with parameters θ_i is randomly initialised. The student model $\mathcal{S}$ starts with parameters θ_c identical to $\mathcal{T}_c$ and is updated during the unlearning phase.

Unlearning Process: The unlearning process is initiated by exposing the student model $\mathcal{S}$ to both teachers. The knowledge from $\mathcal{T}_i$ is meant to corrupt the student's understanding of the target data $\mathcal{D}_f$, which is to be forgotten, whereas $\mathcal{T}_c$ ensures the retention of correct information for the remaining data $\mathcal{D}_r$. This dual influence is governed by the following loss function for each data sample x:

$$L(x, l_u) = (1 - l_u) \cdot KL(\mathcal{T}_c(x) \| \mathcal{S}(x)) + l_u \cdot (KL(\mathcal{T}_i(x) \| \mathcal{S}(x)))$$

Here, KL represents the Kullback-Leibler divergence, measuring how much two probability distributions are distant, and l_u is the unlearning label, indicating whether a sample belongs to $\mathcal{D}_f$ ($l_u = 1$) or $\mathcal{D}_r$ ($l_u = 0$). The goal is to minimize this loss over the dataset, effectively causing $\mathcal{S}$ to unlearn $\mathcal{D}_f$ while retaining predictions similar to the original model $\mathcal{T}_c$ for $\mathcal{D}_r$.

The student's parameter update is guided by the gradient of L concerning θ_c, optimized via stochastic gradient descent.

This approach uses the different knowledge from $\mathcal{T}_c$ and $\mathcal{T}_i$ to selectively influence the student model's learning trajectory. This helps the model forget specific information while maintaining its overall predictive abilities.

2.1.2 Scrub (SCalable Remembering and Unlearning unBound)

SCRUB (SCalable Remembering and Unlearning unBound) [10] is an improved version of the teacher-student model. Instead of just copying the teacher model's

predictions, SCRUB trains the student model to ignore the teacher's output for data that needs to be removed. This selective ignoring creates high error rates for the removed data, which helps eliminate biases and correct mislabeled data. This selective ignoring allows for intentionally high error rates on the unlearned data, which is particularly advantageous for eliminating biases or correcting the impacts of mislabeled data. Additionally, SCRUB uses a 'rewinding' technique to find the best point for unlearning. This technique carefully determines the optimal unlearning checkpoint to minimize potential Membership Inference Attacks (MIAs). However, since MIAs are typically used to evaluate model privacy risks, using them as a target could introduce potential issues. Despite this, the methods in SCRUB together maintain the model's utility while improving data forgetting quality.

2.1.3 Fine-tuning

Fine-tuning as an unlearning method involves fine-tuning the neural network on the data that should be retained while excluding the data that needs to be forgotten. This approach adapts the model parameters θ by minimizing the loss on the retained dataset $\mathcal{D}_r$ using the existing trained parameters as the starting point. The fine-tuning process effectively adjusts θ using the gradient descent method, $\theta' = \theta - \eta \nabla_\theta L(\theta, x)$, where η is the learning rate and $\nabla_\theta L(\theta, x)$ represents the gradient of the loss function. By concentrating the training loop on $\mathcal{D}_r$ and ignoring $\mathcal{D}_f$, fine-tuning aims to reduce the influence of the data that have to be forgotten on the model's performance.

2.1.4 NegGrad (Negative Gradient Ascent)

NegGrad, or Negative Gradient Ascent, is an unlearning algorithm designed to induce forgetting by modifying the gradient descent process typically used in training neural networks. In contrast to traditional approaches that minimize the loss function, NegGrad employs gradient ascent to gradually increase the loss associated with the data points designated for forgetting. This is implemented by adjusting the model parameters θ in the direction that maximizes the loss, $\theta' = \theta + \eta \nabla_\theta L(\theta, x_f)$, where η is the learning rate and $\nabla_\theta L(\theta, x)$ is the gradient of the loss function. By heightening the loss for specific data points in the dataset $\mathcal{D}_f$, NegGrad degrades the model's performance on these points to try to forget the information connected to these data and reduce its influence on the model's overall predictive behavior.

2.1.5 Advanced NegGrad with Classification Loss

Advanced NegGrad is an enhanced version of the original NegGrad approach proposed in [11], designed to optimize the unlearning process through a more sophisticated manipulation of the gradient ascent technique. Unlike basic NegGrad, which focuses only on increasing the loss of the data to be forgotten, Advanced NegGrad integrates a joint loss function that balances retaining useful information and forgetting specific data. This algorithm adjusts the model parameters θ using the formulation $\theta' = \theta + s\eta\nabla_\theta L(\theta, x)$, where $s = 1$ for $\mathcal{D}_f$ and $s = -1$ for $\mathcal{D}_r$. In this way, the gradient is computed not only to maximize the loss on $\mathcal{D}_f$ but also to minimize the loss on $\mathcal{D}_r$. Advanced NegGrad aims to refine the unlearning process, ensuring that the model forgets the targeted data without significantly compromising its overall performance on the remaining data maintaining the model utility.

2.1.6 CF-k (Class-wise Forgetting)

CF-k [12], standing for 'Catastrophic Forgetting-k layers', is an unlearning method where the model's last k layers are incrementally trained on the retained dataset $\mathcal{D}_r$ while other layers remain frozen. CF-k is designed to efficiently erase the influence of the forget set $\mathcal{D}_f$ by focusing the retraining process on the final layers, where higher-level, more dataset-specific representations are typically learned. This targeted unlearning reduces computational costs and minimizes the impact on the overall model performance compared to retraining all layers. Moreover, CF-k provides a flexible trade-off between unlearning efficiency and the depth of erasure by adjusting k, thereby allowing a controlled forgetting process based on the sensitivity of the data and model architecture specifics.

2.1.7 UNSIR (Unlearning by Selective Impair and Repair)

This method [13] incorporates an innovative two-step process: the Impair Phase and the Repair Phase, designed to selectively manipulate network weights to forget specific data and only then retain overall model performance. This method can be used both to forget specific samples or entire classes.

1. Impair phase

In the Impair Phase, UNSIR uses an error-maximizing noise matrix to worsen the model's performance on the target data classes that need to be forgotten. This noise matrix is generated from the original model's predictions and is used to perturb the model weights significantly. In this way, the model's weights are adjusted to

sharply deteriorate its accuracy on the forget data D_f, effectively 'impairing' the model's ability to recall or recognize these data points.

2. Repair phase

Following the initial disruption in the Impair Phase, the Repair Phase is implemented to stabilize the model and restore its previous performance on the remaining data classes. This phase involves a more conservative learning rate and focuses on fine-tuning the model using only the retained data D_r. The objective is to 'repair' any damage to the model's accuracy caused by the Impair Phase, ensuring that the model retains its utility for the remaining tasks.

The combined effect of these phases allows UNSIR to efficiently unlearn specific information while maintaining the model's overall integrity and accuracy.

2.2 Metrics

Metrics are essential for quantitatively assessing whether a machine learning model has successfully 'forgotten' specified data points. This ensures that the model's performance and behavior align with the desired unlearning objectives. Using appropriate metrics provides a standardized framework for comparing different unlearning methods, facilitating the identification of the most effective approaches for various applications.

Some metrics, also used in the section 5, are described below.

2.2.1 Model's Utility

One of the primary metrics used for evaluating unlearning methods is **Unlearning Accuracy** [14], which measures how much the updated model $\mathcal{M}'$ approximates the performance of a model trained solely on the remaining dataset $\mathcal{D}_r$. High unlearning accuracy indicates that the influence of the unlearned data points $\mathcal{D}_u$ has been successfully removed. Another critical metric is **Performance Degradation**, which assesses the change in the model's performance (e.g., accuracy, precision, recall) after the unlearning process. This metric ensures that the unlearning method does not compromise the model's performance.

2.2.2 Method's Efficiency

Unlearning Efficiency [14] is also a crucial metric, evaluating the computational and time cost associated with the unlearning process. This includes the time

complexity and resource utilization required to achieve unlearning. Efficient methods are, in fact, vital in large-scale applications where retraining from scratch is impractical.

2.2.3 Membership Inference Attacks (MIA)

The currently most widely used and recognized metric to assess the unlearning efficiency of certain data is Membership Inference Attack (MIA) [10] [15] [16].

MIA is used to evaluate the effectiveness of Machine Unlearning techniques since it can reveal whether a specific data point was part of the training dataset or not. Recent research [17] [18] has shown that it may be possible with MIA to infer with high accuracy whether an example was used to train a model. For this reason, by employing MIAs as a metric, it is possible to assess whether the influence of these data points has been successfully eliminated. Thus, MIAs provide a quantifiable measure of the residual information the model retains about the unlearned data.

Consider an originally trained model $\mathcal{M}$ trained on dataset $\mathcal{D}$. Typically, the loss values for data points x in $\mathcal{D}$ are lower than those for unseen data points x_{test} from $\mathcal{D}_{\text{test}}$. An MIA identifies whether specific data have been used to train a machine learning model by analyzing the confidence in the predictions (usually the model's loss function). To do this, MIA tries to predict whether an example is part of the forget set $\mathcal{D}_f$ or an unseen set $\mathcal{D}_{\text{test}}$. The more it cannot recognize the difference between the two data sets, the more influential the unlearning method has been.

2.2.4 Forgetting Score

The forgetting score [11] measures the degree of forgetting essential to ensure that the model no longer retains information about the unlearned data. To evaluate the forgetting performance, we utilize an MIA, training an additional binary classification model $\psi(\cdot)$ to distinguish between the loss values of data points that were part of the training set $\mathcal{D}_{\text{forget}}$ and those that were not. The binary classification model $\psi(\cdot)$ is defined as:

$$\psi(x) = M = \begin{cases} 1 & \text{if } x \in \mathcal{D}_{\text{forget}} \\ 0 & \text{if } x \in \mathcal{D}_{\text{test}} \end{cases}$$

If the accuracy M of $\psi(\cdot)$ is 0.5, the machine unlearning algorithm perfectly forgets, indicating that the data points $x \in \mathcal{D}_{\text{forget}}$ are indistinguishable from those in $\mathcal{D}_{\text{test}}$. Finally, the forgetting score is given by $|M - 0.5|$, where a lower score indicates better unlearning performance. For the forgetting score metric values closer to 0.5 are preferable.

Chapter 3

Method proposed

The method described here is based on the approach outlined in Section 2.1.1, with some changes to better use the available data. This optimization aims to effectively optimize the use of forget data in the Forget Set $\mathcal{D}_f$ while preserving the information related to the Retain Set $\mathcal{D}_r$. As evidenced in Section 5, these adjustments have successfully reduced the susceptibility to Membership Inference Attacks (MIA) without significantly extending the duration of the Unlearning phase or degrading the model's performance.

3.1 Data Selection

The initial phase of the unlearning algorithm focuses on data preparation and selection, which is crucial for the success of the subsequent unlearning process.

3.1.1 Similarity-Based Sampling (SBS)

In the unlearning phase, not all the images of the Retain Set $\mathcal{D}_r$ help reconstruct the student model, and this not only slows down the unlearning process but mitigates it as the dimension of $\mathcal{D}_r$ is much greater than the $\mathcal{D}_f$. The objective of the Retain Set is to reconstruct the feature space around the Forget images after the 'destruction phase', as shown in section [13]. This work aims to do so by using Retain images close to that feature space. A graphic example of such an idea is shown in Fig. 3.2.

The methodology involves two key steps, explained in the following sections

Feature Extraction

Let X_{retain} and X_{forget} represent the sets of input data from which features are to be retained and forgotten, respectively. The feature extraction process transforms

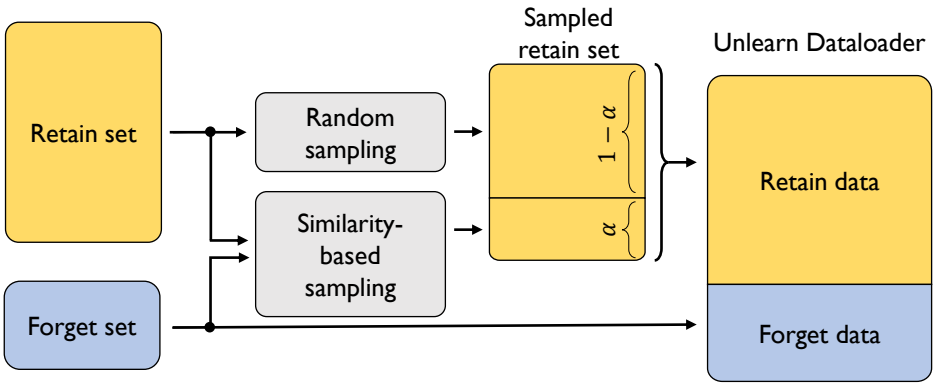

Figure 3.1: Diagram of the Data Preparation for Unleraning. α is the *'smart fraction'*, i.e., the fraction of images sampled based on the similarity of the new Retain subset, described in Section 3.1.1.

these inputs into feature vectors, F_{retain} and F_{forget}, through a feature extraction model M, such that:

$$F_{\text{retain}} = M(X_{\text{retain}}), \quad F_{\text{forget}} = M(X_{\text{forget}})$$

M operates by removing the head of the Competent Teacher Model to focus on layers dedicated to feature extraction. In this way, a proxy of the information in the latent space of the images is made.

The correlation between features in F_{retain} and F_{forget} is quantified using cosine similarity, defined as:

$$S_{\cos}(a, b) = \frac{a \cdot b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

for vectors a and b. This results in a correlation matrix C, where C_{ij} represents the similarity between the i-th feature in F_{retain} and the j-th feature in F_{forget}.

Image Selection

Based on the correlation matrix C, features in F_{forget} that exhibit a high degree of similarity to those in F_{retain} are identified. To do this, the top k elements with the highest cosine similarity in the matrix are considered. This means that the images considered may repeat and that the number of retained images taken is not homogeneous over the number of forgotten images, as shown in Fig. 3.2. This

approach is particularly relevant in scenarios where an individual requests data deletion, as it is common for the training dataset to contain multiple images of the same subject closely clustered in the feature space. Therefore, a retained image, $\mathbf{x}_r$, might share characteristics with a significant subset of $\mathcal{D}_f$, enclosing many, potentially all, images associated with the deletion request.

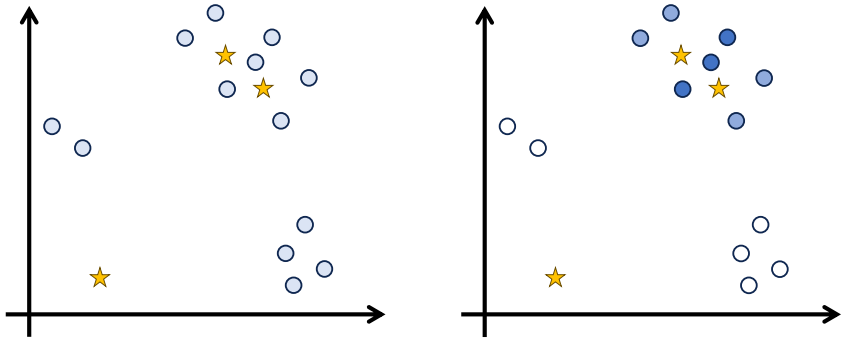

Figure 3.2: Example of image selection for the model reconstruction phase. Stars indicate data points of $\mathcal{D}_f$ in the feature space, and circles represent data points of $\mathcal{D}_r$. Circle shading intensity correlates with the frequency of use in the reconstruction phase. The left image shows the conventional selection method applied in [9], while the right image displays the selection method proposed.

This behavior is justified by the initial assumption made, whereby there may be data present in a more 'dense' feature space, where it will be necessary to take multiple images to reconstruct that feature space in a suitable way considering the absence of the data, and data in more isolated feature spaces (given that the Unlearning framework can also be used for outliers' elimination [19]) it will be sufficient to forget the latter without having to intensively reconstruct the space around it.

In addition, a hybrid solution can be approached, in which only a part of the Retain set $\mathcal{D}_r$ is chosen through this method, and another is chosen randomly from the images that were not selected in this first step to maintain an overall reconstruction.

3.2 Unlearning Process

The second phase of the unlearning algorithm focuses on intelligently using available data to optimize the destruction of information related to points that must be forgotten while effectively restoring the model's usability with the points retained.

3.2.1 Smart Batch Construction (SBC)

Current Limitations

A limitation of the methods based on competent and incompetent teaching, such as the 'Competent and Incompetent Teachers' [9] and 'SCRUB' [10], is intrinsic in the loss function used. These methods minimize the distance of the logits of the retained data between the student model and the competent teacher – that at the beginning of the unlearning phase are equals – and the logits of the data to be forgotten with the incompetent teacher. Considering that in the unlearning framework, the size of the retained set is much greater than that of forget, as visible in Section 4.1, one may run the risk of having many steps having no or minimal effect in the unlearning of the model. Considering the standard method of Competent and Incompetent Teachers, our model will not update until at least one 'forget data' is used in the training loop.

Proposed Solution

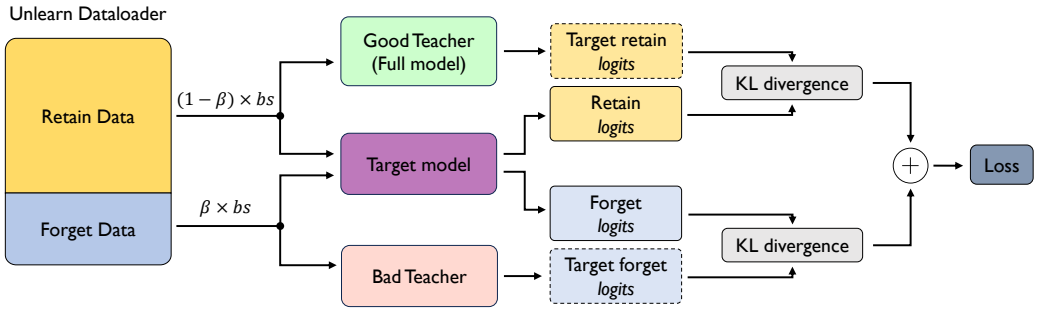

Figure 3.3: Diagram of the Unlearning process. β sets the ratio of Retain and Forget data to take in each step. bs represents the batch size dimension.

In the proposed method, to ensure greater efficiency of the Unlearning process within the epochs, the batch is composed in such a way as to maximize the effect of the Forget Set $\mathcal{D}_f$ and the Incompetent Teacher. As already anticipated, retained points are much more numerous than have to be forgotten, so the effect on the model of the latter could decrease if the batches are not intelligently constructed. For this reason, a process like the one carried out by other unlearning processes like 'UNSIR' (described in Section 2.1.7) of destruction and reconstruction was chosen. At the beginning of unlearning, many more points of $\mathcal{D}_f$ are shown so that the information related to them is destroyed by the Incompetent Teacher. Only then the Competent Teacher reconstruct the model capability with the retained points

in the second phase of the unlearning. This approach is particularly effective at the beginning of unlearning, where the student would produce the same logits as the Competent Teacher on the retained points, which is why the KL-divergence between these two will be zero, making the unlearning phase inefficient.

The parameter β is introduced for this scope. This parameter represents the ratio of points of the Forget Set $\mathcal{D}_f$ with respect to $\mathcal{D}_r$ selected for making a batch.

The approach is delineated through a Smart Batch Construction (SBC), in which the batches for each unlearning step are dynamically composed, initially favoring a higher proportion of Forget Set $\mathcal{D}_f$ data ($\beta = 0.9$) to intensify the unlearning effect. This proportion is adjusted as the process advances, reducing the emphasis on the Forget Set to allow for the reinforcement of the Retain Set knowledge $\mathcal{D}_r$. This adaptive learning process ensures that the Student Model is exposed to a strategically varied learning environment, initially focusing on unlearning (through the Forget Set $\mathcal{D}_f$) and gradually shifting towards re-solidifying its knowledge base using the Retain Set $\mathcal{D}_r$.

Chapter 4

Datasets

The following chapter will outline the formulation of datasets within the general Machine Unlearning framework (4.1) and show the different datasets proposed (4.2) for evaluating the efficacy of the unlearning methods through the definition of a benchmark.

4.1 Dataset Division in MU Framework

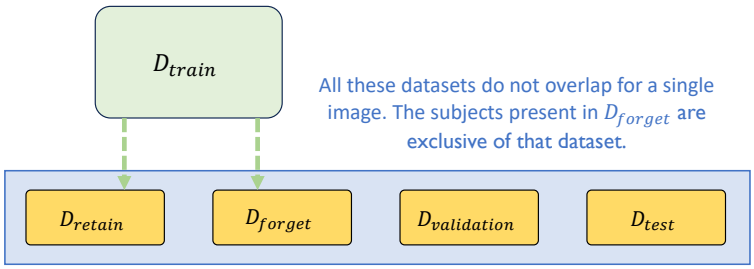

Figure 4.1: Diagram of the data division convention in the unlearning framework

In Machine Unlearning, the Training Set $\mathcal{D}$ is divided into two key subsets to facilitate targeted unlearning: Retain Set $\mathcal{D}_r$ and Forget Set $\mathcal{D}_f$. Additionally, independent validation and test datasets, $\mathcal{D}_{val}$ and $\mathcal{D}_{test}$, are used. These are distinct from $\mathcal{D}$ and are used to evaluate the model's performance and the effectiveness of the unlearning process.

Training Set The Training Set $\mathcal{D}$ in a Machine Unlearning framework is divided into two subsets to facilitate the selective unlearning of data:

- **Retain Set**: This subset comprises data points that must remain within the model's knowledge base. These data points are not targeted for unlearning and are used to ensure that the model retains its ability to perform tasks relevant to these data. At the same time, the unlearning process is applied to other data points.
- **Forget Set**: This subset includes the data points to be forgotten by the model. These are specifically targeted during unlearning to erase their influence from the model's parameters.

The Training Set, $\mathcal{D}$, is exclusively composed of $\mathcal{D}_r$ and $\mathcal{D}_f$, ensuring that every piece of data used for training the model is accounted for in either retaining knowledge or being forgotten.

Validation and Test Sets The Validation $\mathcal{D}_{val}$ and Test $\mathcal{D}_{test}$ Sets are crucial for evaluating the model's performance. They consist of entirely new data that the model has never seen. This distinction is critical for assessments such as Membership Inference Attacks (MIA), where the model's behavior on $\mathcal{D}f$) must be compared against its behavior with data never seen before during the model's training to determine how effectively it has unlearned these specific data.

Exclusivity and Non-overlap: It is vital that $\mathcal{D}_r$ and $\mathcal{D}_f$ are mutually exclusive and collectively exhaustive of $\mathcal{D}$. Similarly, $\mathcal{D}_{val}$ and $\mathcal{D}_{test}$ must not overlap with each other nor with any subset of $\mathcal{D}$. Formally, we define:

$$\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f, \quad \mathcal{D}_r \cap \mathcal{D}_f = \emptyset,$$

$$\mathcal{D} \cap \mathcal{D}_{val} = \emptyset, \quad \mathcal{D} \cap \mathcal{D}_{test} = \emptyset, \quad \mathcal{D}_{val} \cap \mathcal{D} = \emptyset$$

.

This structured approach to Set division supports the methodological integrity of the unlearning process, ensuring that the model's unlearning can be rigorously tested and quantitatively assessed.

4.2 Datasets Proposed

This section is dedicated to presenting the three distinct datasets proposed to test the efficacy of the proposed Machine Unlearning method. The datasets have been

specifically developed to assess the robustness and versatility of the unlearning process across varied data types and complexity levels.

The first two datasets are centered on human facial imagery. In this domain, unlearning frameworks are often most employed due to the sensitive nature of facial data and its extensive use in various applications spanning security, personal identification, and social media. These datasets are inspired by and partially derived from the frameworks suggested in [11].

The third dataset is the CIFAR-100 dataset, which comprises 100 classes encompassing various objects from everyday scenes. This dataset is known for its diversity and complexity, making it a standard benchmark in machine learning for evaluating generalization and performance across generic object classes.

Each of these datasets has been specifically chosen to show the effectiveness of the unlearning method under different scenarios, ranging from sensitive personal data to more general object recognition tasks. The following subsections will describe the composition, specific characteristics, and rationale behind the selection of each dataset, providing a comprehensive basis for the subsequent evaluations made in Section 5.

4.2.1 Forget Set construction

In the structured datasets employed for unlearning experiments, a ratio of 2.5% of the total Training Set $\mathcal{D}$ is designated as the Forget Set $\mathcal{D}_f$. This proportion is maintained to emulate realistic scenarios where a minimal yet significant data segment must be forgotten, typically under privacy deletion requests. This setup tests the model's ability to selectively forget without retraining and aligns with regulatory frameworks emphasizing the Right to be Forgotten [1], providing a realistic benchmark for evaluating the efficacy of unlearning methods under conditions that might be encountered in real-world applications.

4.2.2 Models Training

From all the proposed datasets, two model configurations are employed:

- **General Model:** Trained on the combined Retain $\mathcal{D}_r$ and Forget $\mathcal{D}_f$ sets. This model assesses the initial performance across all available data before any unlearning process.

- **Gold Model:** Trained exclusively on the Retain set $\mathcal{D}_r$. This model serves as a benchmark for the highest expected performance obtainable. In fact, the Machine Unlearning framework aims to obtain a model as similar as possible to the Gold one without retraining everything from scratch.

Both models undergo the same training regimen, described in detail where needed in the following paragraphs for each dataset, ensuring that comparisons reflect differences in data handling and unlearning efficacy rather than variations in model training.

4.2.3 MUCelebA

CelebA [20] is a widely recognized dataset in machine learning. It is mainly used in developing and benchmarking algorithms focused on facial attribute recognition and face detection tasks. It comprises over 200,000 images, each annotated with 40 attribute labels and 5 landmark locations, making it one of the richest datasets available for facial analysis. In our research, we have chosen to simplify the complexity of the problem by focusing exclusively on a single label, 'Arched Eyebrows'. Limiting the scope to one attribute enhances our ability to analyze and refine the model's performance, ensuring a focused and manageable framework for our experimental evaluations. This simplification can isolate the effects of different unlearning strategies on the model's attribute recognition capabilities. An example of images of the MUCelebA dataset is shown in Fig. 4.2

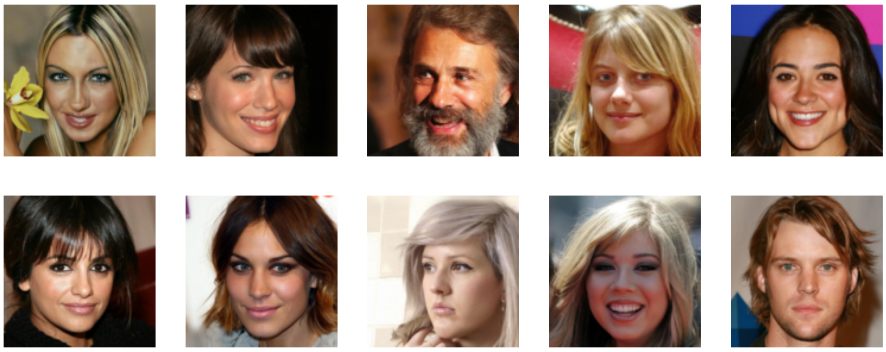

Figure 4.2: Examples of images taken from the Forget Set of the MUCelebA dataset

Dataset Preparation and Division

To develop the unlearning dataset MUCelebA, the CelebA dataset [20] is used. To construct the Forget set $\mathcal{D}_f$, only celebrities with at least 20 images are considered to ensure a robust dataset. These identities are sampled until the $\mathcal{D}_f$ comprises 2.5% of the entire Train Set $\mathcal{D}$, aiming to simulate realistic scenarios of data removal under privacy constraints. This methodological approach should ensure fair model training and practical unlearning experiments across diverse identity representations.

After creating the Forget Set $\mathcal{D}_f$, the remainder of the data is segmented into Retain, Validation, and Test Sets at ratios of 80%, 10%, and 10%, respectively, excluding the $\mathcal{D}_f$ images. The guidelines explained in 4.1 are followed.

Model Training and Data Augmentation

In the models' training phase, a series of data augmentation techniques are systematically applied to increase the diversity of the dataset, prevent overfitting, and simulate a real-world scenario.

The images are first resized to a uniform dimension of 128x128 pixels to standardize input size across all data. Random horizontal flipping introduces variability in the dataset, simulating different orientations and perspectives. Additionally, random affine transformations are applied, which include slight rotations, translations, scaling by a factor of 0.8 to 1.2, and shearing by up to 10 degrees. This helps the model learn to recognize features under various geometric transformations. Color jittering is also applied to adjust the brightness, contrast, and saturation of the images by up to 20%, further enhancing the model's robustness to different lighting conditions and color variations.

These transformations are compiled into a transformation pipeline using a composition of functions, ensuring each image in the training set undergoes the same sequence of transformations (but applied with random parameters every time) before being converted into a tensor for model training.

Implementation and Use

Structured files organize the datasets to facilitate easy access during the machine learning workflows, supporting reproducible results in training, validation, and testing phases across different experimental setups. The models trained on these datasets are also saved to ensure consistency and coherence in the results across various tests. This practice allows for maintaining specific model states, enabling reliable comparisons and evaluations of unlearning effectiveness in subsequent experiments.

4.2.4 Modified MUFAC (MMUFAC)

The MUFAC dataset, introduced in [11], is designed to evaluate Machine Unlearning methods focusing on facial age classification. This novel dataset consists of over 13,000 facial images collected from participants in South Korea, each annotated with age and personal identity number. The age classification process is organized into nine distinct bins, each representing a specific range of age groups.

This setup aims to unlearn specific personal privacy instances while preserving the model's original functionality. Such a configuration ensures that the dataset can

effectively test the robustness and performance of machine unlearning algorithms in realistic and practical settings. An example of images of the MMUFAC dataset is shown in Fig. 4.3

Figure 4.3: Examples of images taken from the Forget Set of the Modified MUFAC dataset

Dataset Preparation and Division

To create Modified MUFAC, the dataset proposed in [11] was examined, and some modifications were applied. In fact, within the dataset, some of the images were duplicated even in the training dataset, making the results obtained from the latter not rigorous for the framework proposed in Machine Unlearning, described also in 4.1.

The first step in generating a new dataset from the one just proposed was to delete all duplicate images within the original dataset. All images with a cosine similarity of 0 to at least another image were removed to do this. Thus, 4,956 out of the initial 10,025 were removed from the Training Set $\mathcal{D}$, 340 out of 1,539 for Validation $\mathcal{D}_{val}$, and 388 out of 1,504 for Test $\mathcal{D}_{test}$. At the end of this first phase, from the initial total dataset of 13,068 images, a dataset of 9,119 unique images was obtained.

After obtaining this new dataset, the images were stratified between $\mathcal{D}$, $\mathcal{D}_{val}$, and $\mathcal{D}_{test}$, resulting in 8,229 training images, 445 validation and test images. The split between Retain $\mathcal{D}_r$ and Forget $\mathcal{D}_f$ was applied following 4.2.1, thus going to select 2.5% of $\mathcal{D}$ for $\mathcal{D}_f$ selecting only identities with at least 16 images. Furthermore, as expressed in 4.1, identities that must be forgotten are absent in any other dataset.

Model Training, Data Augmentation, Implementation and Use

The transformations applied to the dataset images and the related data augmentation are the same as proposed for MUCelebA in Section 4.2.3. Similarly, the data

obtained were saved for reproducibility, as in Section 4.2.3.

4.2.5 MUCifar-100

The CIFAR-100 dataset [21] is an established benchmark in machine learning, mainly used for evaluating image recognition algorithms. Comprising 60,000 32x32 color images, CIFAR-100 is divided into 100 classes, each containing 600 images. What sets CIFAR-100 apart is its organization into 20 superclasses, each encompassing 5 semantically related classes. This structure allows for a layered approach to classification tasks beyond identifying individual classes; models can be trained and tested to recognize and categorize images according to these broader superclass categories. This superclass classification introduces complexity and realism to the task, simulating more nuanced real-world scenarios where objects must be identified individually and to larger categorical groupings. Currently, only the standard classification of the 100 classes is evaluated. An example of images of the MUCifar-100 dataset is shown in Fig. 4.4

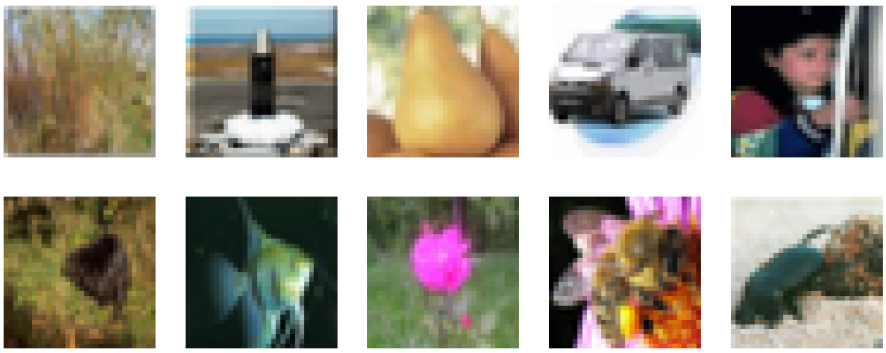

Figure 4.4: Examples of images taken from the Forget Set of the MUCifar-100 dataset

Dataset Preparation and Division

The CIFAR-100 Dataset for each class offers 500 Training images and 100 Testing images. Considering the following division, for MUCifar-100, the images are divided as follows:

- For **Train**, it uses the entire CIFAR-100 Train Dataset.
- For **Retain** and **Forget**, following the guidelines explained in Section 4.1, 97.5% of the images are allocated for the Retain Set $\mathcal{D}_r$ and 2.5% for the Forget Set $\mathcal{D}_f$, guided by a fixed seed.

- For **Validation** and **Test**, the original CIFAR-100 Test Dataset into two halves creating Validation Set $\mathcal{D}_{val}$ and Test Set $\mathcal{D}_{test}$, respectively.

Model Training, Data Augmentation, Implementation and Use

No transformations are applied to the dataset images since the lower resolution of the images of the Cifar dataset and not data augmentation is applied differently from the one proposed in Section 4.2.3.

The data obtained were saved for reproducibility, as in Section 4.2.3.

Chapter 5

Experimental Results

5.1 Experimental Setup

This section describes the experimental framework established to evaluate the effectiveness of the proposed Machine Unlearning method. The experimental design uses distinct datasets to rigorously assess these aspects under various scenarios.

5.1.1 Unlearning Models

The experiments used several unlearning methods to establish comprehensive comparative results:

- **Proposed Method (PM):** Described in Chapter 3.
- **Base Model (BM):** Starting model trained with all the training data, using both retain and forget sets.
- **Retrained Model (RM):** Model trained from scratch using only the retain set.
- **Other Methods:** Includes various state-of-the-art unlearning methods from recent literature, such as Competent and Incompetent Teachers Method (CIT), SCRUB, Fine-tuning, Negative Gradient Ascent (NNegGrad), Advanced NegGrad (ANegGrad), CF-k, and UNSIR. All these methods are described in detail in Section 2.1.

5.1.2 Dataset Description

The experiments were conducted on three specific datasets described in Section 4.2:

- **MUCelebA:** A variant of the CelebA dataset, modified to facilitate the testing of unlearning for facial attributes.
- **Modified MUFAC:** This dataset focuses on facial age attributes, taken from an already existing unlearning dataset [11] and adapted for the unlearning framework guidelines described in 4.1.
- **MUCIFAR-100:** Uses CIFAR-100 dataset to examine unlearning across a more comprehensive collection of object categories and a different environment.

Each dataset was tested using a convolutional neural network (CNN) designed for specific attributes or categories. Initially, networks were trained on the complete datasets without any backbone and saved to set a common baseline for performance. The proposed unlearning method was then applied.

5.1.3 Metrics for Evaluation

To measure the efficacy and efficiency of each unlearning method, the following metrics were considered, described more in detail in Section 2.2:

- **Forgetting Score:** Assesses how well the model protects privacy post-unlearning on the forgot data, estimating susceptibility to inference attacks.
- **F1 Score:** Evaluates the accuracy and utility of the model after the unlearning process.
- **Time Efficiency:** Measures the time required for retraining, reflecting the method's practicality for real-world application.

5.2 Comparative Analysis

The results of the experiments provide a comprehensive analysis of the proposed Machine Unlearning method compared to several baseline and advanced methods. The evaluations are based on the Forgetting Score, F1 score, and computational time, measured across three datasets: MUCelebA, Modified MUFAC, and MUCIFAR-100. This section discusses the key findings from each dataset.

For ease of visualization, the following results show the best models that minimized the Forgetting Score, which is considered the most critical metric for evaluating the goodness of an Unlearning algorithm. Specifically, each table will use a color gradient from darkest to lightest green to indicate the top three configurations, with the darkest green representing the best model, followed by a slightly lighter green for the second best and the lightest green highlighting the third.

For each dataset in this study, another table includes results from the Base Model, the best-performing configuration of the Proposed unlearning Method, and the fully Retrained Model. The Base Model provides a reference point or lower bound for assessing the effectiveness of unlearning, while the Retrained Model serves as an upper bound, representing the ideal scenario of model performance after complete retraining. This structured presentation underscores the effectiveness of the proposed method and places its performance within the broader unlearning settings, enabling an evaluation of its practical applicability and limitations.

5.2.1 Results on MUCelebA

As shown in Table 5.1, the proposed unlearning method outperformed known techniques to reduce MIA accuracy, suggesting enhanced privacy preservation. Additionally, the proposed method maintained a high F1 score, indicating a minimal loss in model utility. Excellent results were also achieved when comparing the Proposed Method with the Base and Retrained Model, as visible in Table 5.2. In fact a Forget Score even lower than that obtained with the Retrained Model is achieved in ~1/10th of the time, keeping the overall F1 score competitive.

5.2.2 Results on MMUFAC

In the Modified MUFAC dataset, the Proposed Method demonstrated overall superior performance with respect to other ones considering all the metrics proposed, as detailed in Table 5.3. Although, on the one hand, the best Forget Score was obtained by NegGrad, the latter had an important loss in F1 Score compared with the proposed method (-10%), and the time required was about 10 times longer. Good results were also achieved when comparing the Proposed Method with the Base and Retrained Model, as visible in Table 5.4. In fact a Forget Score comparable with that obtained with the Retrained Model is achieved in ~1/100th of the time. On the other hand, a drop of ~11% in F1 score is obtained. However, we would like to emphasize that the comparison shown in the table is with respect to the method that obtains a minimal Forget Score, which affects the overall performance of the model. However, it is also possible to choose, through tuning the hyperparameters, a configuration that, at the expense of the Forget Score, increases the usability of the model.

5.2.3 Results on Modified CIFAR-100

The modified CIFAR-100 results, presented in Table 5.5, illustrate the method's effectiveness across a broader range of object categories. As observed in Section 5.2.2, NegGrad is the top-performing method again. However, in this instance,

Table 5.1: MUCelebA - Evaluation of Unlearning Methods

Method	Smart	TF	SF	Forget Score	F1	Time
PM	True	0.2	0.3	0.00814 ± 0.00086	0.83473 ± 0.0035	138.09 ± 2.02
PM	False	0.2	-	0.0222 ± 0.00465	0.84105 ± 0.00052	32.28 ± 0.65
PM	True	0.2	0.5	0.00186 ± 0.00125	0.82152 ± 0.00313	144.67 ± 1.13
PM	True	0.2	0.7	0.00915 ± 0.00275	0.80405 ± 0.00577	147.58 ± 2.94
PM	True	0.2	0.9	0.00475 ± 0.00276	0.77696 ± 0.00626	146.21 ± 0.83
PM	True	0.4	0.3	0.00729 ± 0.00198	0.83709 ± 0.00231	148.95 ± 0.74
PM	False	0.4	-	0.02373 ± 0.00858	0.84075 ± 0.00056	58.72 ± 1.3
PM	True	0.4	0.5	0.00203 ± 0.00183	0.8262 ± 0.00696	152.03 ± 0.84
PM	True	0.4	0.7	0.0039 ± 0.00175	0.8122 ± 0.00488	150.20 ± 1.95
PM	True	0.4	0.9	0.00847 ± 0.00054	0.79019 ± 0.00381	145.48 ± 1.13
PM	True	0.6	0.3	0.00864 ± 0.00361	0.83757 ± 0.00226	153.50 ± 3.38
PM	False	0.6	-	0.02254 ± 0.00576	0.8402 ± 0.00047	88.36 ± 1.09
PM	True	0.6	0.5	0.00475 ± 0.00403	0.82752 ± 0.00234	149.99 ± 1.95
PM	True	0.6	0.7	0.00119 ± 0.00127	0.81723 ± 0.00288	149.43 ± 2.63
PM	True	0.6	0.9	0.00458 ± 0.00148	0.80679 ± 0.00261	149.42 ± 1.00
PM	True	0.8	0.3	0.00695 ± 0.00164	0.83637 ± 0.00327	152.75 ± 3.47
PM	False	0.8	-	0.02254 ± 0.00666	0.84067 ± 0.00080	113.94 ± 1.25
PM	True	0.8	0.5	0.00475 ± 0.00243	0.83199 ± 0.00374	152.11 ± 2.97
PM	True	0.8	0.7	0.00136 ± 0.00068	0.81975 ± 0.00371	150.02 ± 0.61
PM	True	0.8	0.9	0.00424 ± 0.00107	0.81298 ± 0.00319	150.02 ± 0.65
CIT	-	-	-	0.02729 ± 0.00736	0.84095 ± 0.00079	137.61 ± 1.83
SCRUB	-	-	-	0.03373 ± 0.00189	0.83794 ± 0.00218	287.74 ± 3.83
Fine-tuning	-	-	-	0.01458 ± 0.00099	0.84015 ± 0.00051	292.67 ± 13.35
NNegGrad	-	-	-	0.02305 ± 0.00136	0.49216 ± 0.0	665.88 ± 50.63
ANegGrad	-	-	-	0.01864 ± 0.0	0.49216 ± 0.0	613.82 ± 46.1
CF-k	-	-	-	0.01492 ± 0.00138	0.84011 ± 0.00129	330.13 ± 25.81
UNSIR 1	-	-	-	0.00441 ± 0.00136	0.61622 ± 0.00793	18.47 ± 1.29
UNSIR 2	-	-	-	0.02576 ± 0.00175	0.80812 ± 0.00084	162.35 ± 6.19

Table 5.2: Comparison between Base Model, Proposed Method, and Retrained Model for the MUCelebA dataset

Models	Forget Score	F1	Time
Base Model	0.03136	0.84249	-
Proposed Method	0.00119	0.81723	~149 s
Retrained Model	0.01864	0.82117	~4.800 s

the method renders the resulting model ineffective, reducing the F1 score of the classification to ~0. This even more pronounced result is probably due to the greater complexity and variance of the different elements to be forgotten, which is the goal

Table 5.3: MMUFAC - Evaluation of Unlearning Methods

Method	Smart	TF	SF	Forget Score	F1	Time
PM	True	0.2	0.3	0.06311 ± 0.00668	0.44954 ± 0.00288	4.80316 ± 0.25547
PM	False	0.2	-	0.07244 ± 0.00333	0.47219 ± 0.0024	1.4332 ± 0.01707
PM	True	0.2	0.5	0.05689 ± 0.00333	0.40916 ± 0.00796	4.58285 ± 0.03898
PM	True	0.2	0.7	0.03289 ± 0.00259	0.35937 ± 0.00368	4.67907 ± 0.18256
PM	True	0.2	0.9	0.02844 ± 0.00259	0.32721 ± 0.01458	4.5801 ± 0.02906
PM	True	0.4	0.3	0.06711 ± 0.00259	0.44776 ± 0.00223	5.47629 ± 0.10373
PM	False	0.4	-	0.072 ± 0.00301	0.46917 ± 0.00443	2.71508 ± 0.03119
PM	True	0.4	0.5	0.05333 ± 0.00243	0.41891 ± 0.00427	5.45323 ± 0.02643
PM	True	0.4	0.7	0.04222 ± 0.00579	0.36493 ± 0.0044	5.55084 ± 0.08248
PM	True	0.4	0.9	0.03067 ± 0.00475	0.34111 ± 0.00217	5.53582 ± 0.05423
PM	True	0.6	0.3	0.06711 ± 0.00788	0.44936 ± 0.00249	6.52877 ± 0.21303
PM	False	0.6	-	0.07244 ± 0.00178	0.46894 ± 0.00499	4.1143 ± 0.11052
PM	True	0.6	0.5	0.06089 ± 0.00178	0.42663 ± 0.00605	6.56231 ± 0.1797
PM	True	0.6	0.7	0.04889 ± 0.00674	0.38279 ± 0.01026	6.58748 ± 0.08568
PM	True	0.6	0.9	0.03289 ± 0.00259	0.35284 ± 0.00538	6.4724 ± 0.07618
PM	True	0.8	0.3	0.072 ± 0.00573	0.45172 ± 0.00256	7.54654 ± 0.10266
PM	False	0.8	-	0.076 ± 0.00089	0.47181 ± 0.00365	5.33966 ± 0.04867
PM	True	0.8	0.5	0.05822 ± 0.00327	0.43097 ± 0.00363	7.55295 ± 0.07466
PM	True	0.8	0.7	0.04667 ± 0.00544	0.39277 ± 0.00885	7.45845 ± 0.09222
PM	True	0.8	0.9	0.03333 ± 0.00372	0.36436 ± 0.00959	7.62963 ± 0.09218
CIT	-	-	-	0.07289 ± 0.00166	0.47231 ± 0.00574	9.50081 ± 1.79894
SCRUB	-	-	-	0.06756 ± 0.00939	0.4766 ± 0.01866	21.07753 ± 6.49913
Finetuning	-	-	-	0.064 ± 0.00871	0.48205 ± 0.01313	18.00045 ± 0.68536
NNegGrad	-	-	-	0.00844 ± 0.00552	0.2419 ± 0.0066	30.95192 ± 5.06902
ANegGrad	-	-	-	0.05467 ± 0.01019	0.42306 ± 0.00582	34.5031 ± 5.90625
CF-k	-	-	-	0.05956 ± 0.00978	0.48845 ± 0.00629	15.58888 ± 0.39335
UNSIR 1	-	-	-	0.05822 ± 0.0086	0.33562 ± 0.0075	1.92166 ± 0.03935
UNSIR 2	-	-	-	0.04222 ± 0.00932	0.42906 ± 0.00762	10.76562 ± 0.44512

Table 5.4: MMUFAC - Evaluation of Base Model, Proposed Method, and Retrained Model

Models	Forget Score	F1	Time
Base Model	0.06667	0.49227	-
Proposed Method	0.02844	0.32721	~5 s
Retrained Model	0.02444	0.44030	~258 s

of the created dataset. The results obtained are similar to those obtained in the Section 5.2.2. In fact, in the configuration where the Forget Score is minimized, a drop in model performance can also be found.

Table 5.5: MUCifar-100 - Evaluation of Unlearning Methods

Method	Smart	TF	SF	Forget Score	F1	Time
PM	True	0.2	0.3	0.07488 ± 0.00111	0.59845 ± 0.00497	9.96114 ± 0.12069
PM	False	0.2	-	0.1012 ± 0.00179	0.66761 ± 0.00195	1.82243 ± 0.05066
PM	True	0.2	0.5	0.06416 ± 0.00369	0.54263 ± 0.00348	9.91257 ± 0.05686
PM	True	0.2	0.7	0.04264 ± 0.00445	0.47305 ± 0.00672	10.05311 ± 0.08404
PM	True	0.2	0.9	0.0224 ± 0.00236	0.36696 ± 0.00559	9.9123 ± 0.06978
PM	True	0.4	0.3	0.0752 ± 0.00406	0.60914 ± 0.00805	11.34402 ± 0.08833
PM	False	0.4	-	0.1024 ± 0.00524	0.6664 ± 0.00142	3.35824 ± 0.03097
PM	True	0.4	0.5	0.06688 ± 0.00406	0.54656 ± 0.00539	11.30217 ± 0.15009
PM	True	0.4	0.7	0.04816 ± 0.00387	0.49028 ± 0.00938	11.25555 ± 0.12928
PM	True	0.4	0.9	0.02376 ± 0.00412	0.38885 ± 0.00734	11.27956 ± 0.12491
PM	True	0.6	0.3	0.07736 ± 0.00363	0.60796 ± 0.00691	12.6978 ± 0.3074
PM	False	0.6	-	0.10032 ± 0.00272	0.66639 ± 0.0013	4.77986 ± 0.11886
PM	True	0.6	0.5	0.0684 ± 0.00219	0.55463 ± 0.00494	12.50386 ± 0.12446
PM	True	0.6	0.7	0.04584 ± 0.00557	0.48919 ± 0.01	12.81051 ± 0.12623
PM	True	0.6	0.9	0.02456 ± 0.00268	0.3927 ± 0.00713	12.75022 ± 0.05671
PM	True	0.8	0.3	0.078 ± 0.00316	0.61658 ± 0.00589	13.95336 ± 0.17268
PM	False	0.8	-	0.09912 ± 0.00326	0.66499 ± 0.00092	6.29831 ± 0.15163
PM	True	0.8	0.5	0.06768 ± 0.00252	0.55725 ± 0.00284	13.99794 ± 0.22447
PM	True	0.8	0.7	0.04616 ± 0.00259	0.49392 ± 0.00331	14.24257 ± 0.21711
PM	True	0.8	0.9	0.02136 ± 0.00149	0.39776 ± 0.00746	13.94258 ± 0.26383
CIT	-	-	-	0.09808 ± 0.00217	0.66291 ± 0.00223	7.30337 ± 0.04171
SCRUB	-	-	-	0.0992 ± 0.00409	0.66401 ± 0.0082	32.77273 ± 0.53816
Finetuning	-	-	-	0.10784 ± 0.00281	0.67924 ± 0.00148	31.25241 ± 0.57818
NNegGrad	-	-	-	0.01104 ± 0.00796	0.0045 ± 0.0	52.63538 ± 0.31323
ANegGrad	-	-	-	0.03736 ± 0.0047	0.54414 ± 0.00563	59.79322 ± 0.15162
CF-k	-	-	-	0.10344 ± 0.00144	0.67209 ± 0.0017	29.17457 ± 0.23693
UNSIR 1	-	-	-	0.0436 ± 0.00503	0.42138 ± 0.01243	2.78836 ± 0.0607
UNSIR2	-	-	-	0.09968 ± 0.00471	0.66349 ± 0.00336	18.06136 ± 0.154

Table 5.6: MUCIFAR-100 - Evaluation of Base Model, Proposed Method, and Retrained Model

Models	Forget Score	F1	Time
Base Model	0.10520	0.66010	-
Proposed Method	0.02136	0.39776	~14 s
Retrained Model	0.00320	0.65290	~420 s

Chapter 6

Conclusions

The work presents a comprehensive investigation into the Machine Unlearning framework, addressing several critical aspects contributing to the theoretical understanding of data privacy in machine learning.

This research contributes threefold: the development of a novel unlearning method, introducing new datasets for benchmarking unlearning techniques, and a thorough survey of existing unlearning methodologies.

The conclusions drawn from this research identify critical areas for future studies and outline the current limitations and possible enhancements. The final part of the conclusions describes these in detail, guiding efforts to improve the robustness and applicability of unlearning methods in more complex and diverse scenarios.

6.1 Summary of Findings

This work has explored the efficacy of various Machine Unlearning methods, emphasizing developing and evaluating a novel unlearning approach. The experiments conducted across multiple datasets—MUCelebA, MMUFAC, and MUCIFAR-100—and metrics—Forget Score, F1 Score, and Time Efficiency—have provided extensive insights into the capabilities and performance of the proposed method compared to existing techniques. Here, we summarize the key findings:

- **Enhanced Privacy Measures:** The proposed unlearning method consistently reduced the effectiveness of Membership Inference Attacks across all tested datasets keeping the model utility and the time constraints. This improvement indicates a significant enhancement in privacy, addressing the critical need for compliance with stringent data protection regulations such as GDPR.
- **Preservation of Model Utility:** Despite the rigorous unlearning processes,

the proposed method successfully maintained an acceptable F1 score, indicating minimal impact on the model's predictive accuracy and utility. This balance is crucial for practical applications where privacy and performance are critical.

- **Computational Efficiency:** The method demonstrated a competitive edge in computational time compared to the other methods. This efficiency makes it viable for real-time applications and large-scale data environments, which are increasingly common in industry settings.

- **Adaptability Across Diverse Data:** The effectiveness of the proposed method across different types of data (facial attributes and object categories) showcases its adaptability, making it a versatile tool.

These findings underscore the potential of the proposed unlearning method to serve as a robust solution for Machine Unlearning challenges, facilitating regulatory compliance and safeguarding user privacy without sacrificing performance. Future research should aim to expand this method's adaptability to other forms of data and revise the current weaknesses of the proposed method highlighted in Section 5.

6.2 Limitations

One of the principal limitations identified in this work concerns establishing a universally accepted and comprehensive framework for Machine Unlearning that is robust enough to be adopted in legislative contexts. While the proposed unlearning method demonstrates substantial improvements in privacy preservation and computational efficiency, there remains a gap in standardizing these approaches to satisfy legal requirements consistently. Current methodologies, including the one developed in this study, often focus on specific datasets or scenarios, which may not universally translate across different legislative environments where data deletion requests must be handled legally. This limitation is critical because legal standards for data privacy, such as those mandated by the GDPR, require verifiable assurance that the data cannot be reconstructed or inferred. The complexity of achieving this level of compliance is compounded in scenarios involving large-scale data or complex model architectures, where unlearning must be executed without compromising the underlying model's integrity or performance. Future improvements should aim at developing more generalized frameworks that can adapt to varying legal standards and are capable of providing empirical evidence to support the compliance of the unlearning processes.

6.3 Future Work

A key direction for future research is the development of more robust and comprehensive metrics to evaluate Machine Unlearning methods. The effectiveness of unlearning techniques is currently measured by a limited set of metrics, such as the Membership Inference Attack (MIA) accuracy and the Model's Utility, which do not fully capture the possible effects of unlearning. These metrics, while helpful, often fail to address the complex dynamics involved in the unlearning process while considering the final model's performance.

Moreover, as machine learning models grow in complexity and are applied across more diverse scenarios, the need for metrics that can effectively measure the impact of unlearning on model stability, scalability, and performance becomes increasingly important. Future efforts should focus on designing metrics that assess the immediate effects of unlearning and evaluate the long-term implications on model behavior and data integrity. These metrics should clarify how unlearning modifies a model's information landscape, ensuring these changes align with legal standards and operational requirements. The advancement of these metrics will play a crucial role in establishing Machine Unlearning as a reliable and legally compliant tool in a world of AI-driven technologies.

In addition to enhancing metrics, there is a significant opportunity to expand the application of unlearning methods beyond Convolutional Neural Networks (CNNs) to include more complex architectures such as Large Language Models (LLMs). LLMs represent a substantial portion of modern AI applications. However, their complex structures and extensive data in the training phase make unlearning particularly challenging to apply and evaluate. Research into unlearning mechanisms for LLMs could broaden the applicability of these techniques and lead to more responsible use of these models.

Additionally, while this work has primarily focused on the privacy aspects of unlearning, the methodology holds promise for addressing other critical issues in machine learning, such as the problem of model bias [19]. Unlearning can be strategically applied to remove biased data from training sets, thus helping to make models more fair. This approach can be particularly beneficial when initial training data inadvertently incorporates biases that reflect and perpetuate societal inequalities. For this reason, Machine Unlearning also aligns with the growing demand for ethical AI, where the ability to correct and adjust AI behavior post-deployment is crucial.

List of Tables

5.1 MUCelebA - Evaluation of Unlearning Methods 30
5.2 Comparison between Base Model, Proposed Method, and Retrained Model for the MUCelebA dataset 30
5.3 MMUFAC - Evaluation of Unlearning Methods 31
5.4 MMUFAC - Evaluation of Base Model, Proposed Method, and Retrained Model 31
5.5 MUCifar-100 - Evaluation of Unlearning Methods 32
5.6 MUCIFAR-100 - Evaluation of Base Model, Proposed Method, and Retrained Model 32

List of Figures

1.1 Anatomy of the Machine Unlearning framework 3
1.2 Anatomy of the SISA Framework [7]. The dataset D is partitioned into multiple shards $(D1, \ldots, Ds)$, each trained on separate models $(M1, \ldots, Ms)$. The red square highlights the specific data shard that needs to be unlearned. Outputs from all models are then aggregated to form the final model output 5

2.1 Anatomy of the Competent and Incompetent Teachers Framework [9] 9

3.1 Diagram of the Data Preparation for Unleraning. α is the *'smart fraction'*, i.e., the fraction of images sampled based on the similarity of the new Retain subset, described in Section 3.1.1. 15
3.2 Example of image selection for the model reconstruction phase. Stars indicate data points of $\mathcal{D}_f$ in the feature space, and circles represent data points of $\mathcal{D}_r$. Circle shading intensity correlates with the frequency of use in the reconstruction phase. The left image shows the conventional selection method applied in [9], while the right image displays the selection method proposed. 16
3.3 Diagram of the Unlearning process. β sets the ratio of Retain and Forget data to take in each step. *bs* represents the batch size dimension. 17

4.1 Diagram of the data division convention in the unlearning framework 19
4.2 Examples of images taken from the Forget Set of the MUCelebA dataset 22
4.3 Examples of images taken from the Forget Set of the Modified MUFAC dataset 24
4.4 Examples of images taken from the Forget Set of the MUCifar-100 dataset 25

A.1 Example of SBS applied to an image to be forgotten in the MUCelebA dataset 39

A.2 Example of SBS applied to an image to be forgotten in the Modified MUFAC dataset 40
A.3 Example of SBS applied to an image to be forgotten in the MUCifar-100 dataset 40

Appendix A

Similarity-Based Sampling examples

This appendix provides a detailed illustration of the Similarity-Based Sampling (SBS) method. The method's efficacy is demonstrated through specific examples from each dataset, where an image taken from the Forget Set is paired with the nearest images in the feature space of the Retain Set. These examples demonstrate the method's effectiveness in identifying and grouping similar data points. By visually representing these relationships, we highlight how the algorithm can cluster related images, reinforcing the robustness and utility of the employed unlearning strategy. This enhances the model's ability to selectively forget without losing significant contextual information.

For each of the following figures, the first image is the target from the Forget Set, intended for unlearning. The five nearest images from the retained set follow the target image, illustrating the close similarity and relevance between the target and retained data. This visual comparison underscores the effectiveness of the similarity-based approach in accurately identifying and grouping similar instances within the datasets, facilitating targeted and efficient unlearning.

Figure A.1: Example of SBS applied to an image to be forgotten in the MUCelebA dataset

Figure A.2: Example of SBS applied to an image to be forgotten in the Modified MUFAC dataset

Figure A.3: Example of SBS applied to an image to be forgotten in the MUCifar-100 dataset

Bibliography

[1] Alessandro Mantelero. «The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'». In: *Computer Law & Security Review* 29.3 (2013), pp. 229–235 (cit. on pp. 1, 6, 21).

[2] Kate Crawford. *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. Yale University Press, 2022 (cit. on p. 1).

[3] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. «Targeted backdoor attacks on deep learning systems using data poisoning». In: *arXiv preprint arXiv:1712.05526* (2017) (cit. on p. 1).

[4] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. *TOFU: A Task of Fictitious Unlearning for LLMs*. 2024. arXiv: 2401.06121 [cs.LG] (cit. on p. 2).

[5] Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. *Machine Unlearning for Image-to-Image Generative Models*. 2024. arXiv: 2402.00351 [cs.LG] (cit. on p. 2).

[6] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. *A Survey of Machine Unlearning*. 2022. arXiv: 2209.02299 [cs.LG] (cit. on pp. 3, 5, 6).

[7] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. «Machine unlearning». In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 141–159 (cit. on pp. 4, 5).

[8] Jamie Hayes, Ilia Shumailov, Eleni Triantafillou, Amr Khalifa, and Nicolas Papernot. *Inexact Unlearning Needs More Careful Evaluations to Avoid a False Sense of Privacy*. 2024. arXiv: 2403.01218 [cs.LG] (cit. on p. 6).

[9] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. *Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks using an Incompetent Teacher*. 2023. arXiv: 2205.08096 [cs.LG] (cit. on pp. 8, 9, 16, 17).

[10] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. «Towards unbounded machine unlearning». In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on pp. 9, 13, 17).

[11] Dasol Choi and Dongbin Na. *Towards Machine Unlearning Benchmarks: Forgetting the Personal Identities in Facial Recognition Systems*. 2023. arXiv: 2311.02240 [cs.CV] (cit. on pp. 11, 13, 21, 23, 24, 28).

[12] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. «Towards adversarial evaluations for inexact machine unlearning». In: *arXiv preprint arXiv:2201.06640* (2022) (cit. on p. 11).

[13] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. «Fast yet effective machine unlearning». In: *IEEE Transactions on Neural Networks and Learning Systems* (2023) (cit. on pp. 11, 14).

[14] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. *Machine Unlearning: A Survey*. 2023. arXiv: 2306.03558 [cs.CR] (cit. on p. 12).

[15] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. «Amnesiac machine learning». In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 13. 2021, pp. 11516–11524 (cit. on p. 13).

[16] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. «When machine unlearning jeopardizes privacy». In: *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*. 2021, pp. 896–911 (cit. on p. 13).

[17] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. «Membership inference attacks from first principles». In: *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2022, pp. 1897–1914 (cit. on p. 13).

[18] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. «Membership Inference Attacks against Machine Learning Models (S&P'17)». In: (2016) (cit. on p. 13).

[19] Ruizhe Chen et al. «Fast model debias with machine unlearning». In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on pp. 16, 35).

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. «Deep Learning Face Attributes in the Wild». In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015 (cit. on p. 22).

[21] Alex Krizhevsky, Geoffrey Hinton, et al. «Learning multiple layers of features from tiny images». In: (2009) (cit. on p. 25).