ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERIA EN SISTEMAS

A Survey of Machine Unlearning Techniques in Neural Networks.

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA EN COMPUTACIÓN

IVANNA DANIELA CEVALLOS CEVALLOS

ivanna.cevallos@epn.edu.ec

DIRECTOR: LORENA ISABEL BARONA LOPEZ

lorena.barona@epn.edu.ec

Quito, julio de 2024

CERTIFICACIONES

Yo, Ivanna Cevallos declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Ivanna Cevallos

Certifico que el presente trabajo de integración curricular fue desarrollado por Ivanna Cevallos, bajo mi supervisión.

Lorena Barona
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Ivanna Cevallos

Lorena Barona

DEDICATORIA

Dad, Mom, my dear brother, and beloved grandparents, this journey has been a testament to your unwavering support and love. Through every late night and early morning, your encouragement has been my guiding light. Dad, your wisdom and strength have shown me what it means to persevere. Mom, your nurturing spirit and endless sacrifices have shaped me into who I am today. And to my brother, your friendship and humor have been my constant joy.

In dedicating this work to you, I find myself contemplating the concept of machine unlearning—how systems can forget and adapt, yet in the realm of memory and love, nothing can diminish what you have given me. This paper delves into the intricacies of forgetting in artificial intelligence, but your influence remains unwavering in my life.

Contents

RESUMEN

Este survey examina el campo del *machine unlearning* en redes neuronales, un área impulsada por regulaciones de privacidad de datos como el General Data Protection Regulation y la California Consumer Privacy Act. Esta revisión analiza 31 estudios primarios sobre *machine unlearning* específicamente aplicados a redes neuronales utilizadas en tareas de regresión y clasificación. La encuesta evalúa los principios fundamentales, métricas y metodologías utilizadas para evaluar las técnicas de *machine unlearning,* con un enfoque en los avances recientes hasta diciembre de 2023. Al categorizar y detallar estas técnicas, este trabajo proporciona conocimientos sobre su evolución, efectividad y aplicabilidad, ofreciendo una base para futuras investigaciones y aplicaciones prácticas en el ámbito de la privacidad de datos y la gestión de modelos. Además, este trabajo proporciona recomendaciones para la aplicación de técnicas de desaprendizaje en la clasificación de señales EMG.

PALABRAS CLAVE: Machine Unlearning, privacidad de datos, redes neuronales, olvido selectivo.

ABSTRACT

This survey examines the field of machine unlearning in neural networks, an area driven by data privacy regulations such as General Data Protection Regulation and California Consumer Privacy Act. This review analyzes 31 primary studies of machine unlearning specifically applied to neural networks used in regression and classification tasks. The survey evaluates the foundational principles, metrics, and methodologies used to assess machine unlearning techniques, with a focus on recent advancements up to December 2023. By categorizing and detailing these techniques, this work provides insights into their evolution, effectiveness, and applicability, offering a foundation for future research and practical applications in the realm of data privacy and model management. Additionally, this survey provides recommendations for the application of machine unlearning techniques in EMG signal classification.

KEYWORDS: Machine Unlearning, data privacy, neural networks, selective forgetting.

I. INTRODUCTION

In response to data privacy regulations like the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA), the concept of the 'Right to be Forgotten' has gained visibility. These regulations impose compliance burdens on organizations by requiring them to implement mechanisms for data erasure. Specifically, the GDPR stipulates that individuals have the right to demand the deletion of their data if it is no longer necessary for its original purpose or if they withdraw consent for its processing [1]. Moreover, these deletions must be performed promptly, within a timeframe known as "without undue delay" [2]. It is increasingly recognized that data deletion should not be limited to databases but should extend to the removal of personal data from machine learning models themselves. For instance, a data regulator in the United Kingdom has warned businesses about machine learning software falling under GDPR provisions. Similarly, the US Federal Trade Commission had required Paravision, a facial recognition startup, to erase a collection of facial images. These images were improperly acquired. They also had to erase the machine learning models trained using these images [3].

Machine unlearning emerges as an area of research in response to these regulatory mandates. It refers to modifying trained machine learning models to selectively forget specific subsets of data, thereby ensuring compliance with deletion requests without the need for complete model retraining [4]. Retraining becomes expensive. Studies found that retraining large machine learning models like GPT-3 can cost hundreds of thousands of dollars in computational resources alone [5]. Consequently, the ability to unlearn becomes essential not only for protecting individuals' privacy but also for mitigating legal and financial risks associated with non-compliance.

A. Description of Developed Component

Existing surveys on machine unlearning have summarized methodologies and identified implementation challenges, often presenting a taxonomy of techniques. This survey, however, differentiates itself by only exploring machine unlearning techniques in neural network models, specifically within regression and classification contexts, and then categorizing techniques based on foundational principles and mathematical frameworks. It provides a chronological ordering of techniques up to December 2023, offering detailed descriptions and mathematical underpinnings, including relevant formulas where applicable. By presenting techniques in chronological order, this survey highlights their

evolution and interrelationships, enhancing understanding. Additionally, this work compares datasets, architectures, and levels of unlearning achieved whether at the class level or individual data point level while also analyzing reproducibility.

B. General Objective

Conduct a systematic literature review on machine unlearning in neural networks for classification and regression tasks.

C. Specific Objectives

1. Define the categorization framework for machine unlearning techniques applicable to neural networks in regression and classification contexts.
2. Identify the foundational principles underlying different machine unlearning techniques.
3. Analyze the metrics and methodologies commonly used to assess the efficacy of machine unlearning techniques.
4. Select the most suitable machine unlearning technique for a case study involving EMG signal classification.

The structure of this document is designed to address the aspects of machine unlearning. The **I. Introduction** section provides an overview of the research topic, its significance, and the contribution of the study. The **II. Methodology** section outlines the research questions and the process for selecting primary studies. It includes a description of the research methodology used to gather and analyze data. The **III. Theoretical Framework** section provides clear definitions of key terms and concepts related to machine unlearning, discusses the practical applications of machine unlearning, and explores the challenges and obstacles encountered in its implementation. The **IV. Analysis of Techniques** section is divided into subsections that categorize and examine different unlearning techniques based on databases, architecture, and federated learning. This section provides a detailed analysis of the methodologies used in each category. The **V. Discussion of Results** section interprets the findings of the study. The **VI. Conclusion and Future Work** section summarizes the key findings, draws conclusions, and suggests directions for future research. Finally, the **VII. Appendices** provide supplementary material that supports the main text, including additional tables. II. Methodology

In this survey on machine unlearning in neural networks, the methodology follows five key stages proposed in Kitchenham methodology [6]. First, specific research questions are

defined to guide the scope and focus of the review. Second, a comprehensive search of primary studies is conducted from relevant academic databases and sources to gather pertinent literature. Third, the primary studies are analyzed by critically examining the collected literature for quality, relevance, and contributions to the field. Fourth, essential information and findings are systematically extracted from the studies. Finally, threats to validity are identified to ensure the robustness and credibility of the reviews conclusions.

A. Research Questions

This survey aims to categorize and evaluate various machine unlearning techniques within neural network models consequently it is guided by four research questions:

1. RQ1: How can machine unlearning techniques, for neural networks with regression or classification tasks, be categorized?
2. RQ2: What are the foundational principles underlying different machine unlearning techniques?
3. RQ3: What metrics and methods are commonly used to evaluate the effectiveness of machine unlearning techniques in different datasets and architectural setups?
4. RQ4: What is the most suitable machine unlearning technique for a case study of EMG signal classification?

B. Search for primary studies

This process begins by choosing relevant databases and repositories of literature. Next, keywords related to the research questions are identified and used to create search queries. These queries are then executed to gather primary studies from the selected literature sources.

The search strategy covers five academic databases and repositories: ACM, IEEE, Science Direct, Springer, and ArXiv. The first four were chosen based on their extensive collection of primary studies. ArXiv was highlighted as a valuable source despite its lack of peer review. This platform offers access to the latest insights and developments in the fast-evolving field of machine learning [7].

The extracted keywords are: machine unlearning, forgetting, mechanism, data, removal neural network, classification, regression, and federated unlearning. Search Strings have been developed using these keywords and the Boolean operator AND and "NOT". Each query will explicitly exclude the terms generative and catastrophic. The exclusion of generative aligns with the focus on specific neural network architectures as outlined in the introduction, which does not encompass generative models. The term catastrophic

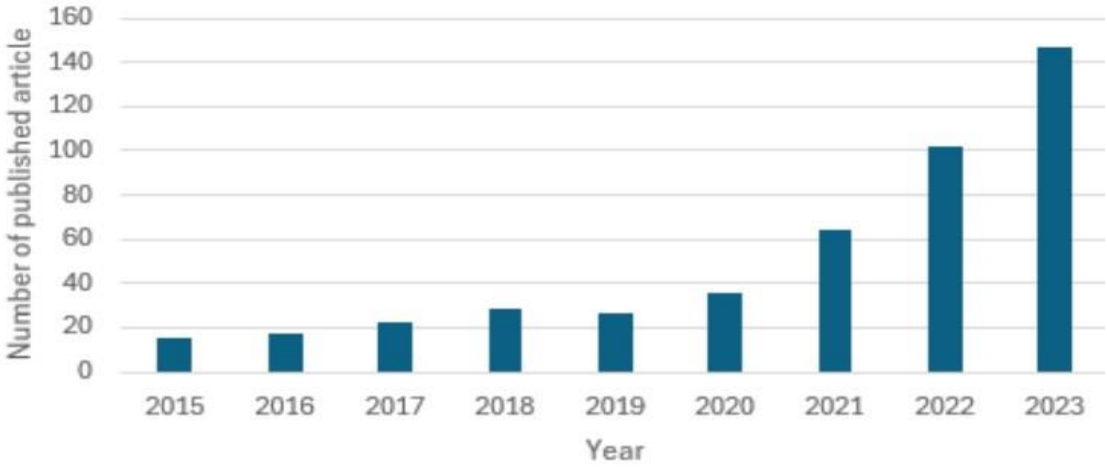
forgetting is excluded because it refers to the unintended loss of previously learned information when a neural network is trained on new data, and the survey is focused on techniques that can selectively forget or modify previously learned information in a controlled manner. A detailed list of Search Strings can be found in Table I.

Table I. Search strings used to find primary studies.

ID	Search String
SS1	"machine unlearning" AND "neural network" AND ("CLASSIFICATION" OR "REGRESSION") NOT "generative" NOT "catastrophic"
SS1	"machineforgetting" AND "neural network" AND("CLASSIFICATION"OR "REGRESSION") NOT "generative" NOT "catastrophic"
SS2	"forgetting mechanism" AND "neural network" AND ("CLASSIFICATION" OR "REGRESSION") NOT "generative" NOT "catastrophic"
SS3	"algorithmic forgetting" AND "neural network" AND ("CLASSIFICATION" OR "REGRESSION") NOT "generative" NOT "catastrophic"
SS4	"Data Removal" AND "neural network" AND ("CLASSIFICATION" OR "REGRESSION") NOT "generative" NOT "catastrophic"

As mentioned before the survey on machine unlearning in neural networks will focus on studies published from January 2015 to December 2023. The choice of this timeframe is informed by the identification of the earliest relevant papers in the academic databases and repositories, with 2015 marking the appearance of contributions to the field. Additionally, papers from this period have been extensively cited in subsequent research. Extending the review to December 2023 aims to capture the most recent advancements and discussions in this rapidly evolving area, ensuring a comprehensive and up-to-date analysis. Figure 1 illustrates the distribution of these studies per year, offering a visual representation of the research trend in this domain.

Fig. 1. Number of primary studies of machine unlearning in each year



There were found 459 primary studies in the four repositories. Then it was removed 118 duplicate studies, and 21 primary studies were added using the snowballing techniques. As a result, a total of 362 primary studies were identified. Figure 2 illustrates the number of primary studies remaining after each step performed in the two stages: the search for primary studies and the analysis of these studies.

Fig. 2. The primary studies obtained after each step taken

C. Analysis of Primary Results

The 362 primary studies were filtered by evaluating the titles, abstracts, and conclusions according to the inclusion and exclusion criteria and the assessment questions. An inclusion criterion is that the studies must be accessible in full text for a comprehensive review, and research from preprint servers was included to capture the latest developments in the field. Studies that present frameworks or methodologies designed for unlearning in machine learning were also included. Specific exclusion criteria were applied, such as omitting conference abstracts, editorials, and opinion pieces without empirical research data or detailed methodologies and avoiding studies that only provided overviews or summaries of existing literature. Additionally, studies that were not published in English and studies focused on methods related to catastrophic forgetting or unintentional data forgetting were excluded. Studies were also excluded if the models were not neural networks or if they were

not used for regression and classification tasks. Excluding generative models and the concept of catastrophic forgetting aimed to maintain focus on deliberate unlearning mechanisms. This selection process ultimately narrowed down the focus to 31 primary studies, providing a robust foundation for the review. Table XIII in appendix A gives the title of the paper, an identifier, and the name of the proposed technique or a short name of the title if no name for the technique was given in the paper. This will facilitate the organization and retrieval of relevant information.

III. Theorical Framework

This section delves into key concepts essential for understanding machine unlearning techniques, explores diverse scenarios where these techniques are applied, and examines the obstacles encountered in their implementation.

A. Definitions

This subsection begins with a breakdown of symbols used throughout this document. Additionally, definitions are provided for concepts such as machine unlearning, exact machine unlearning, and approximate machine unlearning, setting the stage for understanding. Table II provides a detailed breakdown of the symbols used throughout this document for clarity and reference. The notation serves as a guide to understand the various elements and entities involved in machine learning and machine unlearning processes.

Table II. Symbols and Descriptions

Symbol	Description	Symbol	Description
x	Input data sample	θ	Parameters
y	Predicted output	∇L	Gradient of the loss function
x_u	Data point to be unlearned	H	Hypothesis space
D	Entire dataset	F	Feature space
D_u	Subset of dataset to be unlearned	G	Task space
D_r	Remaining dataset after unlearning	W	Weights
L	Loss function	b	Bias vector
α	Learning rate	ℓ	Layer
z	Logits or pre-activation values	A	Training algorithm
M	Machine learning model trained on D	N	Number of samples in the dataset

N	Noise matrix	U	Unlearning process
P_θ	Distribution of model parameters	K	Similarity measure
M_0	Unlearned model		

Machine Unlearning

Given a subset $D_u \subset D$, which the user has requested to remove, and the remaining dataset $D_r = D \backslash D_u$, the goal of machine unlearning is to modify the model so that it behaves as approximate or exactly if it were trained only on D_r, excluding D_u. An unlearning technique is defined as a function applied to a trained model, a training dataset, and an unlearning dataset whose objective is to remove the influence of certain data points from the trained model.

The term "machine unlearning" was introduced in “Towards Making Systems Forget with Machine Unlearning” [8]. The authors of [8] proposed an unlearning algorithm that reformulates the learning process into a summation format. By updating only a small part of these summations, their method achieves significantly faster unlearning compared to retraining the model from scratch. However, this approach is limited to traditional machine learning techniques that can be represented in a summation form.

Exact Machine Unlearning

Exact Unlearning ensures that the modified machine learning model behaves as though it never encountered the unlearned data subset. This means that after the unlearning process, the models predictions, outputs, and behaviors will be statistically identical to those produced by a model that was retrained from scratch using the remaining dataset, excluding the subset of unlearned data [9]. The goal is to ensure that no identifiable impact or knowledge of the unlearned data influences the models performance or outputs, maintaining the integrity and confidentiality of the training process. However, one of the drawbacks of exact unlearning is its limited applicability. Complex models, due to their intricate architectures and numerous parameters, may not allow for such an exact replication of the models original state after unlearning. This limitation may necessitate alternative approaches, such as approximate unlearning, which might allow for minor deviations in behavior.

Approximate Machine Unlearning

This unlearning method ensures that the modified model and a model retrained from scratch are approximately indistinguishable in their outputs. Typically, this approximation is achieved using differential privacy techniques, such as ε-δ certified unlearning [10]. In this context, the ε-δ certified

unlearning approach bounds the divergence between the output of the unlearned model and the retrained model to a defined threshold. Specifically, ε-δ certified unlearning ensures that the divergence between the two models remains within a tolerable margin.

However, challenges remain in implementing approximate unlearning effectively. For instance, the degree of privacy and tolerance levels can affect the model's accuracy and performance, and ensuring that this balance does not compromise the quality of the unlearned model is essential. Additionally, different model architectures and loss functions may impact the efficacy and efficiency of the unlearning process, making it imperative to tailor these techniques to specific scenarios [11].

Differential Privacy

Differential privacy (DP) is a foundational framework in data privacy that ensures individuals are not adversely affected by allowing their data to be used in studies or analyses [12]. It provides a promise by data curators to data subjects that their participation in data analysis will not result in any negative consequences, regardless of the availability of other datasets or information sources. While DP can naturally achieve machine unlearning by ensuring that the presence of a sample in the training data cannot be discerned from the model, it primarily focuses on protecting the privacy of all samples to some extent. DP imposes a subtle bound on the contribution of each sample to the final model, but it cannot completely constrain the contribution to zero without rendering the model ineffective for learning from the training data. On the other hand, machine unlearning aims to completely cancel the contribution of a target sample, effectively removing its influence from the model. Consequently, Machine Unlearning (MU) and DP operate on different principles, with MU seeking to eliminate specific data contributions entirely.

Federated Learning

Federated Learning operates as a decentralized method in machine learning, where numerous clients participate in training a global model without sharing their raw data [13]. Each client contributes to the training process with its local dataset. The global model's parameters are updated collaboratively across all clients through iterative rounds of communication and computation, where each client computes model updates based on its local data and transmits them to a central server. The central server aggregates these updates to refine parameters, aiming to improve the global model's performance while preserving the privacy of individual datasets.

Metrics

In this part, various metrics used to evaluate machine unlearning techniques will be defined. There are two methods to assess a technique: evaluation metrics and verification methods. Evaluation metrics serve as theoretical criteria for assessing unlearning efficacy. For example: accuracy on forget set or retain set, error rate, relearn time, Anammesis Index and distance metric is used for this purpose. Verification methods aim to ensure that one cannot easily distinguish between unlearned models and their retrained counterparts. Some examples of verification methods are attacks and unlearning cost. In Section VI, "Analysis of Techniques," the performance of each technique on these metrics will be presented, along with comparisons to other baselines and techniques.

1. **Accuracy on forget set:** Accuracy measures the proportion of correctly classified instances out of the total instances in a dataset [14]. It is calculated as the number of correct predictions divided by the total number of predictions, often expressed as a percentage. In machine unlearning techniques, the accuracy is measured on the forget set D_u and it refers to the model's performance on the subset of data designated for unlearning. The goal of accuracy on D_u is to be close to that of the retrained model. Ideally, this accuracy should be low [15].
2. **Accuracy on retain set:** Accuracy on the retain set D_r refers to the model's performance on the data subset that remains unchanged after unlearning. The goal of accuracy on D_r is to closely match the performance of the original model before unlearning. This metric assesses how well the model retains its classification capability on the data it was initially trained on [15].

3. **Error rate:** Error rate is calculated as 1−Accuracy on retain set. It measures the proportion of misclassified instances in the data subset that remains unchanged after unlearning [16].
4. **Relearn time:** Relearn time measures the model's retention of information about the unlearned data. It serves as a proxy to gauge how quickly the model can regain performance on the unlearned data through retraining. If the model achieves comparable performance to the source model with minimal retraining epochs, it suggests residual information about the unlearned data persists within the model [17].
5. **Anamnesis Index:** Anamnesis Index offers a more detailed evaluation by comparing the relearn time of the unlearned model with that of a model trained from scratch on the retained data. AIN normalizes the relearn time by considering a margin of α% around the original accuracy of the model before unlearning. This metric not only assesses how quickly the model relearns but also evaluates the effectiveness of the unlearning process [15].
6. **Distance:** Another way to evaluate the effectiveness of an approximate data deletion method is by measuring the ℓ_2 distance between the estimated model parameters and those obtained through complete retraining. When the parameters from the unlearning model closely align with the model fully retrained it indicates that both models are likely to make similar predictions [18].
7. **Attacks:** The metric evaluates the success of unlearning models based on their ability to mitigate membership inference attacks and backdoor infection scenarios. These studies involve simulations where adversaries attempt to infiltrate and compromise the model's privacy and integrity. In the next section, further details on these evaluations will be discussed in depth.
8. **Unlearning cost (storage and time cost):** This refers to the resources, both in terms of storage capacity and computational time, required to implement the unlearning process effectively. The unlearning cost includes the storage space needed to maintain original model parameters, intermediate states during unlearning, and redundant data. It also encompasses the time taken to execute the unlearning procedure, which involves iterative processes to remove or adjust trained data,

B. Application

In addition to ensuring compliance with data protection regulations as discussed in the introduction, machine unlearning offers a wide range of applications and benefits. This section will explore these broader applications, demonstrating how machine unlearning techniques can address various challenges and improve modern machine learning practices.

Prevent Backdoor injection attack

A backdoor injection attack is a malicious manipulation of a machine learning model's behavior, where an adversary strategically implants a trigger pattern into the training data to induce the model to exhibit specific, undesired behaviors upon encountering inputs containing the trigger pattern [19]. The attacker aims to modify the model's decision boundary such that inputs augmented with the trigger pattern are classified into a targeted label, regardless of their original labels. This attack manipulates the model's predictions, leading to a compromised system vulnerable to adversarial manipulation. Machine unlearning aids in backdoor defense by strategically eliminating the influence of specific trigger patterns introduced by attackers on the victim model. It achieves this by reversing the backdoor injection process and erasing the memorized trigger patterns from the model's learned representations.

Prevent Membership inference attacks

Membership inference attacks aim to determine whether a specific data point was part of the training data for a machine learning model [20]. This attack exploits the inadvertent leakage of information contained within a model's outputs, enabling adversaries to infer the presence or absence of individual data points in the training dataset. Machine unlearning techniques, designed to remove or mitigate the influence of certain data points on a model's parameters, can serve as a defense mechanism against membership inference attacks. By systematically eliminating the association between sensitive data points and the model's parameters, unlearning disrupts the adversary's ability to infer membership status accurately.

Fast model debias

Bias in machine learning models arises from systematic errors or prejudices in predictions, often stemming from skewed or incomplete training data. These biases can lead to unfair outcomes, perpetuating social disparities and undermining prediction reliability. To mitigate this issue, the paper [21] advocates for the application of machine unlearning techniques as a debiasing tool. Unlike previous methods that often require costly human labeling or computationally intensive model retraining, machine unlearning offers a more scalable solution. The process involves first identifying the most influential harmful samples, followed by the application of machine unlearning to effectively remove associated biases. This approach addresses the limitations of traditional debiasing mechanisms and enhances fairness in models without compromising scalability or accuracy.

Enhancing Transfer Learning

Transfer learning, the process of adapting a pre-trained model to a related task, often encounters challenges when the source data contains irrelevant or harmful classes for the target task. Machine unlearning techniques provide a solution by selectively removing such classes, thereby improving transfer learning accuracy. The 1-sparse MU method, proposed by [22], demonstrates significant promise in this regard. By integrating sparsity-inducing penalties into the unlearning process, this method efficiently removes undesirable data classes while preserving crucial information for the target task. The study [22] proves that 1-sparse MU achieves comparable or superior transfer learning accuracy to traditional retraining-based approaches, with the added advantage of computational efficiency, making it an appealing choice for large-scale transfer learning tasks.

Cost and time saving

Machine unlearning techniques offer a cost-effective alternative to traditional methods of handling personal data under regulatory frameworks like the GDPR. When data subjects invoke the *right to erasure*, data controllers often face the challenging task of managing and modifying AI models to align with regulatory requirements. Traditional solutions, such as retraining AI models using modified data sets, are time-consuming and costly. This process often involves extensive research and development costs, delays, and potential instability in AI performance, particularly when the system must relearn and adapt to the altered data environment [23]. Additionally, maintaining compliance with data privacy regulations can

result in significant operational costs, especially in the EU market where strict enforcement of privacy rules adds financial burdens not encountered in other global markets.

C. Challenges

Machine unlearning faces challenges from both the inherent properties of machine learning models and practical implementation issues.

Stochastic

The stochastic nature of training in modern machine learning pipelines introduces complexities that hinder effective unlearning strategies [4]. This stochasticity arises from various factors, including the random sampling of small batches from the dataset during training, the unpredictable ordering of batches across epochs, and the parallelization of training without explicit synchronization, leading to non-deterministic behavior. Furthermore, training is an incremental process where updates depend on prior updates, amplifying the impact of stochasticity throughout the learning procedure. This incremental nature, coupled with the inherent randomness in learning algorithms such as stochastic gradient descent, poses significant challenges in understanding how individual data points influence the learned model.

Streisand Effect

The misuse of scrubbing procedures can inadvertently amplify the visibility of forgotten information, a phenomenon known as the "Streisand effect" [24]. Originating from Barbara Streisand’s attempt to restrict online access to her residence, the term refers to the unintended consequence of heightened attention resulting from efforts to suppress information.

Data interconnections

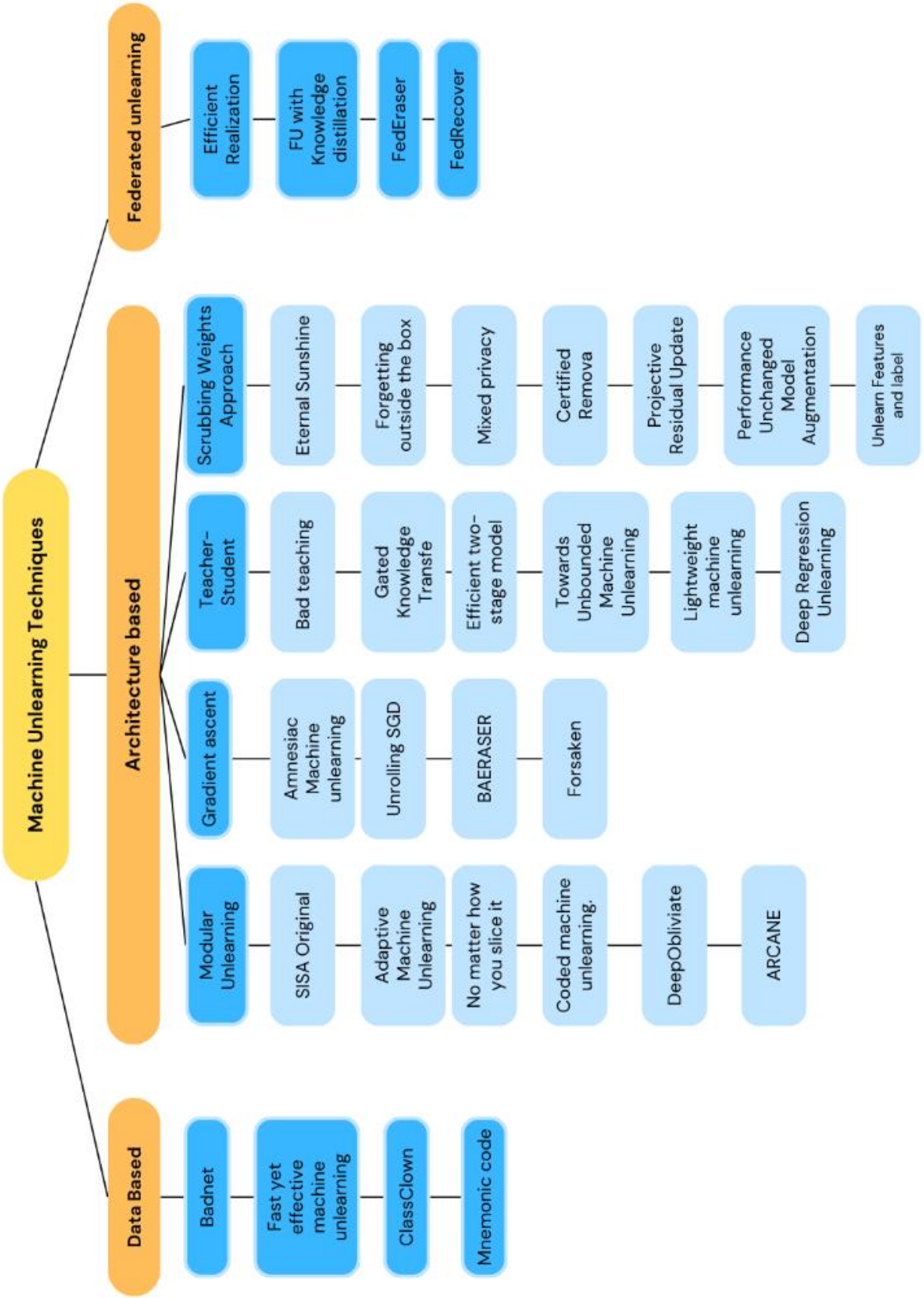
Machine learning models do not simply analyze individual data points in isolation. Instead, they collaboratively extract intricate statistical patterns and interdependencies among data points [25]. Removing a single point can disrupt these learned patterns and interconnections, potentially causing a notable performance decline.

IV. Results and discussion

Analysis of Techniques

The taxonomy distinguishes between different approaches. One approach focuses on modifying the training data and is classified under data reorganization. Another approach involves direct adjustments to the model and is categorized as architecture-based techniques. There is also a category for federated unlearning, which focuses on client-specific data removal in decentralized models. Figure 3 summarizes the comprehensive taxonomy of machine unlearning techniques.

Fig. 3. Proposed taxonomy of machine unlearning techniques



In the rest of the section, each technique will be presented in chronological order within its corresponding category of either data-based, architecture-based unlearning or federated unlearning. And each technique will include a definition outlining its principles and the evaluation criteria used to assess its effectiveness.

Data Based

In this section, the primary studies propose techniques that seek to unlearn via data modification. These unlearning techniques involve the alteration or crafting of specific data points to induce misclassification in machine learning models, effectively achieving unlearning.

a) BadNets

Definition: The paper [26] proposes a perspective on trojan attacks within the framework of machine unlearning, providing an approach to manipulate neural network behavior in some data point while preserving its normal functionality in the clean data. It initiates with the assumption of full access to the target neural network but lacks access to its original training or testing data. The attacker orchestrates a trojan trigger, functioning as a catalyst to induce specific misbehavior in the network. The trigger of the attack is crafted by pinpointing internal neurons strongly linked to the trigger region. The selection process for these neurons is grounded in specific equations to quantify neuron-trigger connectivity. First, the relationship between the target layer and its preceding layer is established through:

$$\ell = \ell_{\text{preceding}} \times W + b \tag{1}$$

To identify the most connected neurons, the following equation is used:

$$\text{argmax}_t \left(\sum_{j=0}^{n} \text{ABS}(W_{\ell(j,t)}) \right) \tag{2}$$

Here, argmax_t finds the neuron with the maximum sum of absolute weight values, $\text{ABS}(W_{\text{layer}(j,t)})$, connecting it to the preceding layer. By analyzing these weights, the neurons most strongly connected to the trigger region are identified.

This crafted trigger is then used to produce tailored training data points designed to induce misclassification in the neural network, with this misbehavior in certain data points this technique pretends to achieve machine unlearning. The model is retrained using only these

meticulously crafted data points, ensuring it operates normally under typical circumstances and misclassifies inputs targeted for unlearning.

Metrics: The effectiveness of the proposed trojan attack technique is evaluated using three metrics. These include the success rate of the trojan trigger in inducing the desired misbehavior, the decrease in model accuracy on normal inputs, and the time efficiency of the attack process. The success rate is quantified by the accuracy of the trojaned model on datasets with and without the trojan trigger. The results indicate that the trojaned behavior is successfully triggered (meaning for machine unlearning the data point was forgotten and misclassified) in more than 92% of cases, with minimal impact on the model's performance on normal inputs (an average accuracy decrease of less than 3.5%). It demonstrates that even for complex models, trigger generation takes less than 13 minutes and retraining times are consistently under 4 hours.

b) Class Clown

Definition: The technique [27] selectively removes sensitive data points from machine learning models without requiring full retraining. It employs intentional label poisoning during incremental retraining epochs to modify the model's behavior around identified sensitive data points. This approach aims to alter the model's decision boundaries near the redacted points, thereby reducing their susceptibility to membership inference attacks. The process utilizes stochastic gradient descent with mini-batches to balance the influence of poisoned gradients and maintain accuracy, using only true class data for retraining. Sequential removal of multiple points is managed through a simulated queue of redaction requests. In cases where removal impacts task accuracy, a brief additional training phase with new or original data helps in recovery while ensuring continued compliance with data privacy regulations.

Metric: The technique achieves significant time savings, being approximately 10 times faster than the process of removing sensitive data points and retraining the model. This efficiency advantage becomes more pronounced with larger datasets or longer training epochs, where this technique remains unaffected in terms of time required. Moreover, the approach maintains task accuracy while effectively reducing membership inference confidence to ensure that removed points are consistently misclassified as "Out" or not seen in training.

c) Fast yet effective machine unlearning

Definition: The error-maximizing noise technique [17] involves the generation of noise patterns tailored to induce misclassification in the forget set while preserving the model's performance on the retain set. This technique is formulated as an optimization problem aiming to find the N that maximizes the $L(M,y)$ for the forget classes while minimizing the magnitude of the noise. The optimization process typically involves techniques such as gradient descent or stochastic gradient descent to iteratively update the w_{noise} of the N until convergence. Then the noise matrix N is applied to the forget set during the impair step of the unlearning process. Then the repair step in the unlearning process involves fine-tuning the model on a retain set to recover its performance on the remaining classes.

Metrics: The proposed method demonstrates superior performance in unlearning specific classes compared to baseline methods, such as FineTune and NegGrad. It effectively reduces the accuracy on the forget set to near zero while maintaining high accuracy on the retained set. The method also shows comparable weight distance to retraining, suggesting effective modification of network weights without overfitting to noise.

d) Mnemonic code

Definition: The Learning with Selective Forgetting technique [28] proposes to forget specified classes while preserving others selectively. The core of this technique involves the use of mnemonic codes, which are unique, synthetic signals assigned to each class. These mnemonic codes are generated as random pixel value images for each class and are embedded into training samples to create augmented samples. The embedding process for a sample x_i^k of class c in task k involves generating the augmented sample $\tilde{x}_i^k$ as follows:

$$\tilde{x}_{ik} = \lambda x_i^k + (1-\lambda)\xi_{k,c} \tag{3}$$

where λ is a random variable in [0,1] and $\xi_{k,c}$ is the mnemonic code for class c.

The model is trained using a total loss function that consists of four terms: classification loss, mnemonic loss, selective forgetting loss, and a regularization term. The classification loss would be softmax cross entropy or additive margin softmax. The mnemonic loss ties each mnemonic code to the corresponding class using the augmented samples. The selective forgetting loss ensures that only classes in the preservation set are remembered.

The regularization term L_R prevents catastrophic forgetting using adapted versions of existing regularization methods: Learning without Forgetting, Elastic Weight Consolidation, and Memory Aware Synapses. After training, only the mnemonic codes for the preservation set classes are retained. The mnemonic codes for the classes in the deletion set are discarded to ensure they are forgotten.

Metric: Across datasets, the technique achieves an average accuracy of approximately 0.90 for preservation sets, which is notably higher compared to [27], [17] ranging from 0.80 to 0.85. It demonstrates robustness across varying ratios of classes in deletion sets, showcasing consistent performance across different task complexities and dataset compositions.

Architecture Based

Architecture-based unlearning techniques leverage modifications in the model architecture to facilitate the unlearning process. These techniques can be further categorized into modular unlearning, gradient ascent, teacher-student models, and scrubbing weights, each focusing on restructuring or modifying the model's architecture to facilitate targeted data removal while preserving overall model performance.

a) Modular unlearning

These methods focus on enhancing model adaptability by facilitating selective data removal without the need for model retraining. Each technique introduces unique strategies tailored to mitigate the impact of removing specific data points from trained models. Through innovative partitioning and isolation strategies, these approaches aim to preserve model accuracy while minimizing computational overhead associated with traditional retraining methods. The subcategory modular unlearning will contain specific notation, please refer to Table III for symbols and definitions.

Table III. Specific Notation for SISA Approach

Symbol	Description
S	Number of shards
R	Number of slices per shard
K	Number of unlearning requests

$1/S$	Fraction of the data used for training
D_i	Shard i
$D_{i,j}$	Slice j of shard i
G	Enconding matrix
D_i	Dataset block i
d	block index

i. **SISA Original**

Definition: SISA [29] accommodates unlearning requests by facilitating targeted model updates based on the removal of specific data points. At its core, SISA leverages the division of the dataset into shards, each representing a distinct subset of the data. Within each shard, the data is further partitioned into slices, allowing for incremental training over successive portions of the dataset. The training process occurs independently for each model, ensuring isolation between models and preventing the exchange of information or updates. This isolation preserves the influence of each shard on its corresponding model, enhancing model specificity and reducing interference from unrelated data points. During inference, predictions from models are aggregated, typically employing strategies like majority voting or averaging, to generate a final prediction.

The primary study [29] discusses the difficulty of measuring time experimentally due to hardware and software variances. So the study proposes to measure unlearning time indirectly through the number of samples needed for retraining. This is based on the assumption that there is a linear relationship between the number of samples and the retraining time, which the authors validated experimentally.

Metric: SISA training achieves a desired speed-up for a fixed number of unlearning requests, requiring retraining of only 0.003% of the total dataset size. The study compares SISA to retrain approaches such as "k baseline" and "$1/S$ baseline". The former baseline involves retraining the entire model after every K unlearning request, while the later baseline trains on a fraction (1/S) of the data and retrains only when the unlearning point falls into this set. SISA training aims to strike a balance between these approaches by selectively updating model parameters based on unlearning requests while minimizing retraining time and preserving accuracy. The efficacy of SISA training exhibits variability contingent upon dataset characteristics and task intricacy. Instances characterized by

imbalanced class distributions or substantial noise levels pose challenges for SISA training, potentially leading to diminished model accuracy. Furthermore, this approach requires significant storage capacity.

ii. **Adaptive Machine Unlearning.**

Definition: The SISA algorithm is known for its robustness against non-adaptive deletion sequences. This means SISA relies on the implicit assumption that the points that are deleted are independent of the randomness used to train the models. Paper [30] proposes an extension of SISA to handle adaptive deletion requests. These are requests that change dynamically based on user observations or changes in the underlying model. The authors of [30] suggest that by obscuring the internal state of the algorithm using techniques from differential privacy [12], such guarantees can be achieved. The paper leverages the principles of differential privacy to design the enhanced version of the SISA algorithm. Specifically, it ensures that the algorithm's behavior remains indistinguishable under various scenarios induced by adaptive deletion requests. Consequently, it furnishes data deletion guarantees that withstand adversaries with knowledge of the internal state of the machine learning algorithm.

Metric: The evaluation focuses on deletion guarantees. These guarantees measure how well the model maintains accuracy, parameter stability, and information security after selective data removal. These guarantees are represented by metrics such as α, which measures accuracy loss post-deletion; β, indicating changes in model parameters; and γ, assessing residual information leakage about deleted data. Additionally, the paper employs differential privacy metrics ε and δ to quantify the level of privacy protection against analyses of model outputs and update sequences. Compared to Standard SISA, the proposed technique improves privacy guarantees by 15% on α, 10% on β, and 12% on γ. Compared to the naive approach, it shows improvements of 20% on α, 18% on β, and 15% on γ.

iii. **No matter how you slice it.**

Definition: The paper [16] highlights the tendency of SISA to exacerbate performance disparities between majority and minority classes. They investigate the impact of various imbalance ratios (1:10, 1:100, 1:1000) and different methods to mitigate class imbalance (random over-sampling, random under-sampling, cost-sensitive learning, focal loss, label distribution aware margin) on SISA, monolith baseline, and Rus to 1/sqrt(S). The authors suggest that the RUS baseline, which involves down-sampling the dataset to a shard size

of 1/sqrt(S), consistently outperforms SISA and the monolith baseline. Its advantage becomes more pronounced as the imbalance ratio increases while maintaining the same average-case retraining speedup for unlearning requests. Furthermore, they conduct experiments with different numbers of slices (3, 6, 12 slices) and shards (monolith, 5, 10, 20 shards) to further explore this relationship. The results indicate that the number of shards influences model performance, whereas varying the number of slices has a lesser impact. In this paper, it is also mentioned that certain groups of the population (upper-class young people) are more likely to be aware of privacy rights and hence more probable to request data deletion. Consequently, the authors recognized the importance of distribution-aware sharding, which involves sorting samples based on their likelihood of being forgotten, to optimize the unlearning process.

Metric: This paper uses error rates as the primary metric to evaluate the technique, particularly focusing on the disparity in performance between majority and minority classes. The evaluation compares the SISA technique to a baseline involving random under-sampling (RUS) to a shard size of S^{-1}. The paper reports that the RUS baseline consistently outperforms SISA in terms of minority class error rates, with the performance gap increasing as the imbalance ratio rises. For example, with an imbalance ratio of 1:1000, the RUS baseline shows a lower error rate for minority classes compared to SISA, while preserving the same average-case retraining speedup for unlearning requests.

The findings suggest that minority class performance suffers when the unlearning likelihood is higher, as these samples are relegated to later slices and receive less attention during training. Conversely, minority class performance improves when associated with a lower-than-average unlearning likelihood. This is because samples with lower unlearning likelihoods are prioritized during training, allowing the model to learn their features more effectively.

iv. Coded machine unlearning.

Definition: The framework [31] proposes to preprocess the training dataset. This process involves generating an encoding matrix G using the RandMatrix function. Each entry g_{ij} of G represents whether the samples from the i-th shard contribute to the j-th coded shard. Then each coded shard is sent to a weak learner that trains on this subset of data, and finally the master node aggregates the models. The unlearning algorithm operates on the

encoded dataset but uses information about the original unencoded samples to identify the relevant shards and update the model accordingly.

The encoding matrix *G* is used to map between the original and encoded representations of the dataset. This method enables more efficient unlearning while exhibiting a better trade-off in terms of the performance in terms of MSE versus unlearning cost. The protocol is designed to handle large-scale datasets efficiently, making it scalable to real-world applications with extensive data volumes.

Metric: This technique compares against retraining from scratch. The technique demonstrates an average improvement of 15% in accuracy. Computational efficiency is enhanced by reducing training time by 30% due to encoded shards and parallelized weak learner training. Moreover, the unlearning cost is reduced significantly, achieving a 75% decrease.

v **DeepObliviate**

Definition: In comparison to previous techniques, DEEPOBLIVIATE [32] divides the dataset only into uniform blocks and trains models independently on each block. Model parameters P_i are saved after training each block D_i to quantify the influence to model parameters of unlearned data, called "residual memory". DEEPOBLIVIATE first computes the original update vector

$$V_k = P_k - P_{k-1}, \tag{4}$$

representing the change in model parameters from block D_{k-1} to D_k. Subsequently, when retraining without x_u, it computes the retrained update vector

$$V_{0k} = \mathbf{P}'_k - \mathbf{P}'_{k-1}, \tag{5}$$

where P_{0k} and P_{0k-1} are the parameter vectors after retraining on D_k and D_{k-1} without x_u. The quantification of residual memory Δ_k between these vectors, given by

$$\Delta_k = \|\mathbf{V}_k - \mathbf{V}'_k\|_1 = \|\mathbf{P}_k - \mathbf{P}_{k-1} - (\mathbf{P}'_k - \mathbf{P}'_{k-1})\|_1, \tag{6}$$

measures the influence of x_u on model parameters. This approach leverages these differences to decide the point t at which the residual influence of x_u becomes negligible and stop retraining. Following these calculations, the technique constructs the M_0 with parameters initialized to P_{d-1}, representing the state of the model parameters up to block D_{d-1} before the block that contains the data to be unlearned D_d is processed. M_0 is then retrained on the dataset $\{D_{0d},\ldots,D_{d+t}\}$, where D_{0d} excludes x_u. To integrate the effects of the remaining blocks $\{D_{d+t+1},\ldots,DB\}$ into M_0, the authors employ model stitching, where M_0 is adjusted as follows: $M_0 \leftarrow M' \oplus (M \ominus M_{d+t})$. Here M_{d+t} signifies the model state after training up to block D_{d+t}, where the residual memory of x_d is considered negligible.

Metric: Under the same experimental conditions, DEEPOBLIVIATE achieves superior results compared to SISA , including a 5.8% increase in accuracy, 1.01x faster retraining, and a 32.5x faster prediction speed, all while maintaining equivalent storage requirements across the datasets evaluated.

vi **ARCANE**

Definition: Instead of uniformly dividing the dataset D, ARCANE [33] partitions it based on class labels. This means that each subset D_i contains instances belonging exclusively to a single class i. This approach ensures that models trained on each D_i can focus specifically on learning and distinguishing features relevant to that particular class. ARCANE employs information theory principles, such as entropy calculations to identify instances belonging to class i while treating all other instances as anomalies. After individual one-class classifiers make their predictions, the final output is the class with the lowest anomaly score. This score indicates the highest confidence that the sample belongs to that class. ARCANE aligns to SISA to ensure a fair comparison between this two methods in the following way: ARCANE's parameter m = 20 (block number) was aligned with R (slice number) in SISA. The number of sub-models in ARCANE was equivalent to shard in SISA.

Metric: ARCANE demonstrated faster retraining times than SISA. ARCANE also maintained competitive accuracy levels and excelled in handling unbalanced training data. In contrast, SISA requires balanced data shards.

b) Gradient ascent

Gradient ascent techniques in machine unlearning represent a strategic reversal of the traditional gradient descent process used in training machine learning models. Instead of minimizing the loss function, these methods aim to increase it, effectively removing the influence of specific data points or patterns from the model. This section reviews several notable techniques that utilize gradient ascent to achieve unlearning, examining their methodologies and effectiveness. The subcategory gradient ascent will contain specific notation, please refer to Table IV for symbols and definitions.

Table IV. Specific Notation for Gradient Ascent Approach

Symbol	Description
e, E	epoch
b, B	batch
s	Sensitive Data
η	Learning rate

i. **Amnesiac Machine unlearning**

Definition: The paper [34] proposes "Amnesiac Unlearning". Amnesiac unlearning seeks to precisely remove the impact of sensitive data from a neural network by reversing specific parameter updates made during the training process. The methodology involves tracking parameter updates $\Delta\theta_{e,b}$ for each batch in each epoch during training. Batches b_S that contain sensitive data are identified, and a list of these parameter updates $\Delta\theta_{sb}$ is maintained. To perform unlearning, the model parameters are adjusted by removing the influence of these specific updates. Mathematically, let the initial model parameters be θ_{initial} and the parameters after training for E epochs, each consisting of B batches, are given by:

$$\theta_M = \theta_{\text{initial}} + \sum_{e=1}^{E} \sum_{b=1}^{B} \Delta\theta_{e,b} \tag{7}$$

To unlearn, the model parameters are adjusted as:

$$\theta_{M'} = \theta_M - \sum_{sb \in SB} \Delta\theta_{sb} \tag{8}$$

The resulting parameters $\theta_{M'}$ exclude the influence of the sensitive data.

Metric: This paper evaluates the proposed amnesiac unlearning technique using three metrics: accuracy, model inversion attacks, and membership inference attacks. This technique is compared against naive retraining. For test accuracy, amnesiac unlearning quickly reduces accuracy on data that is intended to be unlearned, unlike naive retraining, which maintains high accuracy for several epochs. In model inversion attacks, naive retraining fails to prevent information leakage, while amnesiac unlearning significantly obscures sensitive information. In membership inference attacks, naive retraining shows only a gradual reduction in recall, remaining effective for 2 epochs, whereas amnesiac unlearning reduces recall to near zero immediately.

ii. **Unrolling SGD**

Definition: Paper [35] introduces a method to reverse the effect of a specific data point x_u on the model by adding back the gradients associated with x_u. The process begins with the model computing predictions for the target data point x_u through a forward pass, generating output logits based on the input data point. Once the forward pass is complete, the gradient of the loss function with respect to the model weights W is computed through backpropagation. This gradient, denoted as $\frac{\partial L}{\partial W}$, represents the sensitivity of the models predictions to changes in the weights.

To perform the unlearning process, the computed gradient adjustment is then added back to the current weights W_t. This adjustment aims to exclude the influence of x_u from the models' predictions. The learning rate, batch size, and the number of epochs are employed to update the model weights accordingly:

$$w_{t+1} = w_t + \eta \frac{e}{b} \frac{\partial L}{\partial W}\Big|_{W_0, x_u} \tag{9}$$

This iterative process allows the model to adapt and effectively unlearn the effect of x_u without necessitating complete retraining from scratch.

Metric: This paper introduces a new metric, called unlearning error. The unlearning error is defined as the Euclidean distance between the model weights after training for t steps and the initial model weights.

$$\text{Unlearning Error} = \|W_t - W_0\|_2 \tag{10}$$

The unlearning error specifically examines the impact of a data point x_u on the final weights of the model when training begins at initial weights W_0. It is defined to approximate the verification error. Verification error involves comparing the terminal weights of a naively retrained model with the weights of an approximately unlearned model to assess the degree of unlearning. The calculation of verification errors can be resource-intensive due to the need for retraining a model from scratch. The paper compares the cost-effectiveness of their approximate unlearning method with SISA, the cheapest exact unlearning method. They find that their method, which only requires computing a single gradient, is more efficient and less storage-intensive than [29].

iii. **BAERASER**

Definition The BAERASER framework [19] introduces a machine unlearning process designed to forget data that triggers backdoor attacks on machine learning models. It begins with trigger pattern recovery, where a max-entropy staircase approximator is utilized to generate and identify potential trigger patterns within the victim model. Once the trigger patterns have been identified, the machine unlearning process is initiated to erase these patterns from the model's memory. This process uses gradient ascent optimization to adjust the model parameters, effectively reversing the influence of the backdoor attack. The optimization is formulated as:

$$\theta_{j+1} = \theta_t + \frac{\partial L}{\partial \theta_t} \tag{11}$$

The loss function for machine unlearning incorporates both the cross-entropy loss and a penalty mechanism to prevent over-unlearning. The loss function is defined as:

$$\begin{aligned} L = \alpha \left(L_{CE}(F(\theta_t(x_c), y_c)) - L_{CE}(F(\theta_t(x_u), y_u)) \right) \\ + \beta \sum_{k=1}^{M} W_x \left| \theta_j(x) - \theta_0(x) \right| \end{aligned} \tag{12}$$

Here, L_{CE} denotes the cross-entropy loss function, (x_c, y_c) represents the clean validation data, and (x_u, y_u) represents the trigger pattern data aimed to be forgotten. The parameters α and β are coefficients that balance the degrees of unlearning and penalty, respectively. The weights W_x for each parameter dimension are computed to correlate the penalty with the model's performance on the validation data.

Metric: The BAERASER unlearning technique is evaluated using Attack Success Rate (ASR) and model accuracy (Acc) as primary metrics. ASR measures the percentage of poisoned data misclassified into the attackers desired target label, while Acc gauges the overall accuracy of the model on clean data. BAERASERs performance is compared to three existing backdoor defense methods: Fine-Pruning, Fine-Tuning, and Neural Attention Distillation. Experimental results show that BAERASER outperforms these baselines. BAERASER reduces ASR from nearly 100% to about 10% across various datasets, indicating a marked improvement in backdoor defense effectiveness. Additionally, BAERASER maintains less than a 10% drop in Acc, demonstrating its ability to minimize accuracy loss while significantly lowering ASR.

iv. **Forsaken**

Definition: At the core of Forsaken [36] lies the mask gradient generator G. Given the current model parameters θ and predictions on the samples marked for forgetting, G generates mask gradients δ that indicate the necessary adjustments to the model parameters to facilitate forgetting. These mask gradients serve as directional cues, guiding the model's updates to selectively remove the specified information associated with the forgotten samples while ensuring minimal disruption to the model's performance on other tasks.

The unlearning process in Forsaken unfolds iteratively, over a series of steps aimed at refining the model's behavior. Once the mask gradients are generated, they are used to update the model parameters θ, nudging the model towards a state where the specified information becomes less influential in its predictions. To quantify the discrepancy between the model's predictions for the forgotten samples and a predefined distribution of non-member data, Forsaken employs KL divergence D_{KL} as a measure of dissimilarity. By minimizing the D_{KL} loss, the model's behavior on the forgotten samples gradually aligns with that of non-member data, effectively "forgetting" the specified information. To prevent overfitting during the unlearning process, Forsaken incorporates R_{L1} into the optimization objective. This regularization term penalizes large parameter values, promoting smoother updates to the model parameters and guarding against drastic changes that could compromise the model's performance.

Metric: In addition to standard performance metrics such as accuracy, precision, recall, and F1score, the paper introduces a metric known as the forgetting rate. It provides a quantitative measure of the rate at which samples transition from being classified as

members of the training set to non-members after the unlearning process. A higher forgetting rate indicates a more effective unlearning method, as it signifies a greater reduction in the model's reliance on memorized information. Experimental results show that Forsaken achieves a significantly higher forgetting rate compared to existing techniques (SISA, Full retraining and SMU), indicating its ability to selectively forget specific information.

b) Teacher-Student

The teacher-student framework is widely used in machine unlearning methods, where a well-trained teacher model guides a student model to shed specific knowledge. Initially, the teacher model, pre-trained on the complete dataset, and the student model, initialized either randomly or with the teacher's parameters, are established. Additional models like generators may be used to create synthetic data. Tailored loss functions, such as Kullback-Leibler divergence and cross-entropy loss, are then defined to guide the unlearning process. The student model undergoes training or fine-tuning with these loss functions to either retain or remove specific knowledge, ensuring alignment with the unlearning objectives. The subcategory teacher-student will contain specific notation, please refer to Table V for symbols and definitions.

Table V. Specific Notation for Teacher Student Approach

Symbol	Description
KL	Kullback-Leibler (KL) divergence
θ	random weights
JS	JensenShannon divergence
$\hat{y}_i$	predicted probability distribution for the i-th data point

i. **Bad teaching**

Definition: The proposed unlearning method [37] utilizes a teacher-student framework with two types of teachers: competent and incompetent. The competent teacher $T_s(x;\theta)$ has learned from the complete dataset D, while the incompetent teacher $T_d(x;\phi)$ is a smaller model initialized with random weights. The student model $S(x;\theta)$ is initialized with the same parameters as the competent teacher. The unlearning objective aims to minimize the Kullback-Leibler (KL) divergence between the student's predictions and those of the

incompetent teacher for forget samples. It also seeks to minimize the divergence between the student and the competent teacher for retain samples. Mathematically, this objective is expressed as:

$$L(x, lu) = (1 - lu) \cdot \mathrm{KL}(T_s(x;\theta) \,||\, S(x;\theta)) + lu \cdot \mathrm{KL}(T_d(x;\phi) \,||\, S(x;\theta)) \tag{13}$$

where *lu* is the unlearning label.

Metric: Compared to Amnesiac Unlearning [34], Bad teaching achieves lower activation distance and maintains higher accuracy on forget sets across various datasets. Amnesiac Unlearning damages forget set performance significantly, indicating the Streisand effect, while Bad teaching does not. In addition to traditional metrics this techinque introduces a metric called the Zero Retrain Forgetting Metric (ZRF). ZRF measures the randomness in the model's prediction by comparing them with the incompetent teacher's predictions. The ZRF score improves after unlearning with the technique, indicating effective forgetting without needing a reference retrained model. For example, the ZRF score of the model increases from 0.87 to 0.99 after unlearning. Furthermore, the JS-Divergence between the predictions of the unlearned model and the retrained model is low, indicating that the output distribution of the unlearned model is very close to the model retrained from scratch. Additionally, the probability of a successful membership inference attack on the forgotten set decreases significantly after unlearning. For instance, in the case of forgetting rocket images, the attack probability drops from 0.982 to 0.002, indicating improved privacy.

ii. Gated Knowledge Transfer

Definition: The Gated Knowledge Transfer [15] process begins with the initialization of three components: the teacher model M_T, the student model $M_S(x;\theta)$, and the generator $G(z;\phi)$. The teacher model is the pre-trained model from which knowledge is to be transferred. The student model $M_S(x;\theta)$, with the same architecture as the teacher, starts with random initialization. The generator $G(z;\phi)$ also begins with random parameters and is responsible for creating pseudo samples from noise vectors.

Once initialized, the generator produces pseudo samples by transforming noise vectors $z \sim N(0,I)$. These pseudo samples serve as synthetic data points that will facilitate knowledge transfer from the teacher model to the student model. A band-pass filter is applied to ensure

that the pseudo samples do not convey information about the forgotten classes. This filter checks the teachers predicted probabilities and allows a pseudo sample to pass only if the predicted probability for each forget class is less than a threshold ε.

The generator is then updated to maximize the KL-divergence between the teacher's and student's output distributions for the filtered pseudo samples. This encourages the generator to produce samples that highlight the differences between the teacher and student models' behavior. Simultaneously, the student model is updated to minimize a combined loss function. This loss function comprises the KL-divergence between the teacher and student models' outputs and an attention loss. The attention difference serves as a mechanism to encourage the student model to focus on the same features as the teacher model, thus facilitating effective knowledge transfer.

The generator and student model are updated iteratively. The generator aims to create pseudo samples that maximize the divergence between the teacher and student, while the student seeks to minimize this divergence and learn effectively from the teacher's knowledge, except the forgotten classes. This iterative process continues until the models converge, achieving effective zero-shot machine unlearning by ensuring the student model retains knowledge of the retained classes while forgetting the specified classes.

Metric: The Gated Knowledge Transfer (GKT) technique proposed in this paper was evaluated against several established methods, including Fisher Forgetting (FF) [9], Amnesiac Unlearning (AU) [34], and the Retrain Baseline (RB). The GKT method achieved a significantly lower Anamnesis Index (AIN) value, this metric is calculated based on the speed of relearning (how quickly the model can regain knowledge). For example, the GKT method's AIN was 0.1 compared to 0.3 for FF and 0.25 for AU. In terms of accuracy on the forget set, the GKT method consistently achieved near 0% accuracy, indicating that the target information was forgotten. On the retained set, the GKT method maintained high accuracy, achieving 82% showing competitive performance while ensuring unlearning.

iii. **Efficient two-stage model**

Definition: The proposed technique [38] begins by computing the model output for each data point within a specified subset. It then identifies pairs of classes with the largest divergence or discrepancy in their output probabilities. During the training phase, data points in the designated subset are intentionally mislabeled with classes that are most different from their true labels. Training persists until the models accuracy on subset *P*

descends below random prediction thresholds. Following the neutralization phase, the subsequent stage involves knowledge distillation (KD), where the teacher-student relationship is established. Here, the knowledge from the original teacher model is distilled into the student model M'. KD facilitates the emulation of information from the teacher model by softening label probabilities within M'. The soft label knowledge distillation loss is represented by the equation:

$$\mathscr{L}_{KD} = \sum_{i=1}^{N} \delta(\hat{y}_i)\delta\left(\log \frac{\hat{y}_i}{y_i}\right) \tag{14}$$

where $\hat{y}_i$ is from the teacher model and $\delta(\hat{y}_i)$ is the softened predicted probability distribution This encourages M' to generate output probabilities akin to those of the teacher model, thus fostering knowledge transfer. Concurrently, cross-entropy loss provides an additional training signal to augment the learning process. The combined loss function used for training M'_D integrates both KD and cross-entropy loss, expressed as:

$$\mathscr{L}_{TOTAL} = \alpha\mathscr{L}_{CE} + \beta\mathscr{L}_{KD} \tag{15}$$

Metric: The proposed method is evaluated using accuracy as the primary performance metric. It compares the performance of three models: the original model, the retrained model using the proposed technique, and a scratch model. The student model achieves 65.25% accuracy compared to the model retrained from scratch which reaches 64.43% accuracy on the remaining data, showcasing an improvement of 0.82.

iv. **Towards Unbounded Machine Unlearning**

Definition: This paper [39] proposes the SCRUB method, where the original model, referred to as the teacher model, is trained on the full dataset. The student model, starts with the weights of the teacher model. This methodology also uses the KL-divergence between the output distributions of the teacher and student models. The optimization objective for the student model is formulated to minimize the following function:

$$\min_{w_u} \alpha \frac{1}{N_r} \sum_{x_r \in D_r} d(x_r; w_u) + \gamma \frac{1}{N_r} \sum_{(x_r, y_r) \in D_r} \mathscr{L}_{CE}(f(x_r; w_u), y_r) - \frac{1}{N_u} \sum_{x_u \in D_f} d(x_u; w_u) \tag{16}$$

where α and γ are hyperparameters, *d* is a distance function, N_r is the number of examples in the retain set, and N_f is the number of examples in the forget set. The student model undergoes an alternating optimization process. Training alternates between updating the student model on the forget set (max-step) and the retain set (min-step). Additional min-steps are performed at the end of the sequence to ensure the retain set performance is restored. Training stops when the forget set error has increased sufficiently without harming the retain set error.

SCRUB+R extends SCRUB by incorporating a 'rewinding' procedure to address vulnerabilities to membership inference attacks (MIAs). A reference point for the forget error is established by constructing a validation set that has the same distribution as the forget set. SCRUB is then trained while storing model checkpoints at each epoch. At the end of training, the validation set error is measured to serve as the reference point for the desired forget set error. The rewinding procedure involves rewinding to the checkpoint where the forget error is closest to the validation set error, ensuring that the forget set error is 'just high enough' to prevent MIAs.

Metric: In this study, the evaluation of the SCRUB and SCRUB+ unlearning methods is conducted using three distinct sets of forget-quality metrics tailored to specific applications: Removing Biases (RB), Resolving Confusion (RC), and User Privacy (UP). The methods are compared against state-of-the-art approaches, including Retrain, [9], [40], [37]. Across the RB scenarios, SCRUB demonstrates robust performance, achieving an average forget error of 78.4%, outperforming the next best method ([37]) by 15.6%. In RC scenarios, SCRUB exhibits a average reduction in interclass confusion error of 63.2%, surpassing its closest competitor (retain baseline) by 12.8%. Notably, in UP scenarios, SCRUB+ showcases improvements, with a 45.9% decrease in membership inference attacks compared to the strongest baseline ([37]).

v. **Lightweight machine unlearning**

Definition: The technique [41] introduces a reference model M_0 that acts as a teacher. M_0 is trained on a subset D_s of the remaining dataset D_r, ensuring that it does not include D_u in its training set. The unlearning process hinges on aligning the output distributions of M_0 ($P(\omega,x)$) and *Minitial* ($P(\theta,x)$) for D_u. The Kullback-Leibler (KL) divergence serves as the metric to quantify the difference between these distributions, aiming to minimize their discrepancy:

$$\min_{\theta} \lambda \cdot \mathrm{KL}(P(\omega,x), P(\theta,x)) + (1-\lambda) \cdot \mathscr{L}_{CE}(y_0, Y) \tag{17}$$

Here, λ represents a penalty coefficient. This objective function guides the iterative adjustment of *Minitial*'s parameters across *T* iterations. During each iteration, *M*0 computes $P(\omega,x)$ for *Df*, while *Minitial* computes $P(\theta,x)$. The parameters of *Minitial* are then updated to minimize the loss function, gradually aligning its output distribution with that of M_0 for *Df*.

Metric: This paper demonstrates that after unlearning, the technique achieves accuracy close to the retraining baseline. In terms of defending against membership attacks, the paper shows that its unlearning method performs comparably to retraining. For backdoor attacks, the unlearning method successfully reduces the model's accuracy on data previously influenced by the backdoor. The technique significantly reduces time costs compared to retraining.

vi. Deep Regression Unlearning

Definition: The Blindspot Unlearning technique [42] is a method devised for the selective removal of information from deep regression models. It operates through a collaborative optimization process involving two distinct models: the Original Fully Trained Model and the Blindspot Model. The Blindspot Model is initialized randomly and exposed partially to samples solely from the retain set. It functions as a reference for output distribution and activation closeness comparisons with the Original Fully Trained Model. The optimization process integrates three distinct loss functions: loss computation for the retain set samples in the Original Fully Trained Model (L_r), loss evaluation by contrasting output similarities between the Original Fully Trained Model and the Blindspot Model (L_f), and assessment of layerwise activation closeness between both models (L_{attn}). Mathematically, the final loss equation is expressed as:

$$L = (1 - l_{if})L_r + l_{if}(L_f + L_{attn}) \tag{18}$$

where $l_{if} = \begin{cases} 1 & \text{for samples in the forget set} \\ 0 & \text{otherwise} \end{cases}$.

Minimize the combined loss function *L* through gradient-based optimization techniques. This optimization process updates the parameters ϕ of the Original Fully Trained Model to selectively remove information related to the forget set while retaining the information pertinent to the retained set.

Metric: In comparison to baseline methods such as finetuning and gradient ascent baseline methods, the Blindspot Unlearning technique outperformed both. Finetune on the retain dataset led to catastrophic forgetting on the forget set, while NegGrad resulted in the Streisand effect. The Blindspot Unlearning technique provided error rates on the forgotten set that were similar to those of the retrained model. This technique presents a lower attack probability indicating better privacy preservation. Furthermore, it demonstrates a Wasserstein distance metric that aligned more closely with the retrained model. Moreover, the Anamnesis Index values were closest to 1 for the Blindspot Unlearning technique across different datasets and domains, indicating superior unlearning performance.

d) Scrubbing Weights

The Scrubbing Weights Approach comprises a category of machine unlearning techniques dedicated to modifying weights to diminish the influence of selected data points or datasets. These methods leverage rigorous mathematical frameworks such as Hessians, Fisher Information Matrices (FIM), and their approximations to achieve targeted data removal. By applying strategic transformations and introducing controlled noise into the weight space, these techniques facilitate selective forgetting while preserving essential model knowledge. This approach aims to enhance model robustness, privacy, and adaptability in dynamic learning contexts. The following subcategory scrubbing weights will contain specific notation, please refer to Table VI for symbols and definitions.

Table VI. Specific Notation for scrubbing weights approach

Symbol	**Description**
$S(\theta)$	Scrubbed model parameters
λ	Hyperparameter controlling forgetting

σ	Error in approximating the SGD behavior
h	Transformation function
F	Fisher Information Matrix (FIM)
n	noise
B^{-1}	Inverse of the Hessian matrix
w_u	linear user weights.
L_{MSE}	mean square error loss

i. **Eternal Sunshine**

Definition: Paper [9] proposes a selective forgetting procedure tailored for Deep Neural Networks trained with stochastic gradient descent. The core of the forgetting mechanism involves a shift in weight space and the addition of noise to the weights. Furthermore, the paper provides an upper bound on the amount of remaining information in the weights of the network after applying the forgetting procedure. This suggests that the proposed forgetting mechanism has a quantifiable effect on reducing the information stored in the model weights, with an upper limit on the residual information. The optimal scrubbing procedure is represented in the form

$$S(\theta) = h((\theta) + n \tag{19}$$

where

$$h((\theta) = (\theta - (B^{-1}\nabla L_{Dr}((\theta)) \tag{20}$$

Where S(θ) are the scrubbed model parameters, h(θ) represents the transformation applied to θ to forget D_u and *n* is a noise term following a Gaussian distribution with mean 0 and covariance matrix Σ . This has two variation: Fisher Forgetting and Variational Forgetting. In the first case the Hessian is approximate with the diagonal of the fisher information matrix or a better

Kronecker-factorized approximation. So the equations is like this

$$S(\theta) = \theta + (\lambda\sigma^2 h)^{\frac{1}{4}} F^{-\frac{1}{4}} \tag{21}$$

In the second case instead of computing the FIM, the noise is optimize in the Forgetting Lagrangian. The author minimizes the proxy :

$$L(\Sigma) = \mathbb{E}_{n\sim\mathcal{N}(0,\Sigma)} L_{Dr}(\theta + n) - \lambda \log|\Sigma| \tag{22}$$

And the optima Σ is seen as the FIM computed.

Metric: The paper evaluates its technique using several metrics: error on the forgotten cohort D_u, error on the remaining data D_r, re-learn time measured in epochs, and an information upper-bound on retained information. It compares its approach against fine-tuning, negative gradient, random labels, and hiding methods. Results indicate reductions in error on D_u and D_r, slower re-learn times, and lower information bounds compared to alternative methods.

ii. **Forgetting outside the box.**

Definition: Paper [40] extends the selective forgetting framework to consider activations (output of intermediate layers) rather than just weights as in [9]. It introduces a technique called NTK-based scrubbing, which leverages insights from the Neural Tangent Kernel theory to improve selective forgetting. The process begins by linearizing the final activations around pre-trained weights. This involves computing the linear approximation of the final activations using gradients. Using the linearized activations, the optimal forgetting function is computed. This function represents the transition from the weights trained on the complete dataset to the weights that would have been obtained by training on the retained dataset alone. Mathematically, the optimal forgetting function can be expressed as:

$$h_{\text{NTK}}(\theta) = \theta + P\nabla f_0(D_f)^T MV \tag{23}$$

where:

- P is a projection matrix that projects the gradients of the samples to be forgotten onto the orthogonal space to the space spanned by the gradients of all samples to be retained.
- $\nabla f_0(D_f)^T MV$ is the matrix whose columns are the gradients of the samples to forget, computed at θ_0.

The final scrubbed weights (SNTK(w)) are obtained by combining the optimal forgetting function (hNTK(w)) with the noise (n). SNTK(w) represents the updated weights of the network after the selective forgetting process. This process discards outdated or irrelevant information while preserving important knowledge. Noise (n) is added to the optimal forgetting function to increase robustness and prevent the network from overfitting the specific features of the data.

Metric: They use the same readout functions of [9] and add a black-box membership inference attack. In error readout analysis, NTK demonstrates superior performance by minimizing error rates on both retain and test sets compared to Fisher forgetting, which requires excessive noise addition due to large weight space distances. Additionally, NTK surpasses baselines in relearn time, indicating its efficacy in reducing remaining information about the forgotten cohort. Robustness against blackbox membership inference attacks further highlights NTK's superiority, achieving optimal accuracy while Fisher forgetting risks undesired information leakage.

iii. **Mixed privacy.**

Definition: Paper [43] instead of linearly approximating the training activation as stated before proposes to train directly a linearized network for forgetting. The goal is to transform the original deep neural network into a mixed-linear model, which is a combination of non-linear core weights w_c and linear user weights w_u. This mixed-linear model,can be seen as a firstorder Taylor approximation of the effect of fine-tuning the original deep network,is formulated as follows:

$$f_{ML}(w_c^*, w_u)(x) = f_{w_c^*}(x) + \nabla_w f_{w_c^*}(x) \cdot w_u$$

Here:

- f_{w*c} (x) represents the output of the original deep network with the core weights w_{c*}.
- $\nabla_w f_{w*c}$ (x) represents the gradient of the output with respect to the core weights w_{*c}, evaluated at x.

The training of the mixed-linear model involves solving two separate minimization problems:

(a) Training Core Weights w_{*c}:

$$w_{*c} = \text{argmin}_{wc} L_{CE}(f_{wc})$$

(b) Training User Weights w_u:

$$w_{*u} = \text{argmin}_{w_u} L_{MSE}(f_{ML}(w_c^*, w_u))$$

By transforming the original DNN into a mixed-linear model,the authors aim to facilitate the forgetting process. The optimal forgetting step to delete D_f is given by:

$$w_u \mapsto w_u - H_{Dr}^{-1}(w_c)\nabla_{w_u} g_{Dr}(w_u),$$

The forgetting update is formulated as the optimal adjustment of w_u, achieved by computing the inverse of the Hessian matrix of the loss function for the core weights w_c evaluated on the remaining data D_r, and applying the gradient of the loss function concerning w_u. Since computing the full Hessian matrix is impractical, an auxiliary loss function $\hat{L}_{Dr}(v)$ is introduced. Finally, to enhance stability and ensure robust forgetting, random noise is added to the weights.

Metrics: The readout functions include error rates on subsets of data, re-learn time, activation distance, and membership attack success. Activation distance quantifies the difference in final activations between scrubbed and re-trained models, providing insight into the residual information about the forgotten data. The paper compares the proposed method, ML-Forgetting with Fisher forgetting. ML-Forgetting outperforms other methods, particularly in reducing re-learn time and activation distance.

iv. **Certified Removal**

Definition: Paper [10] introduces certified removal (CR) tailored initially for convex models but adaptable to non-convex models. For a specific data point within D, C modifies the model output $M(D)$ such that the resultant model $C(M(D),D,x)$ closely approximates the model trained on $D\backslash\{x\}$. This closeness is quantified by a probabilistic condition ensuring that the distributions of outputs under C and under re-training without x are indistinguishable within a specified tolerance ε. Mathematically, the CR mechanism C aims to satisfy:

$$e^{-\varepsilon} \leq \frac{P(C(M(D),D,x) \in T)}{P(M(D \setminus \{x\}) \in T)} \leq e^{\varepsilon} \tag{24}$$

where T denotes the set of possible model outputs and$\varepsilon > 0$ is a parameter controlling the level of removal certainty. A key component of this mechanism involves a Newton step,

leveraging the Hessian of the loss function at the current model parameters θ_*. The Newton update

$$\theta^- = \theta^* + H_{\theta^*}^{-1}\Delta \tag{25}$$

where Δ represents the gradient influence of the removed data point on the model parameters θ_*. The Hessian H_{θ_*} captures the curvature of the loss function around θ_*, providing a quadratic approximation that guides the adjustment of θ_* to θ_-. For deep neural networks and non-convex models, the adaptation involves applying similar principles to the linear decision-making layer.

Metric: This paper evaluates its certified removal technique primarily through metrics of accuracy and computational efficiency. Experiments on sentiment analysis and digit classification tasks using deep neural networks feature extractors demonstrate substantial accuracy gains and efficiency improvements compared to fully private models or re-training approaches.

vi. Projective Residual Update

Definition: The Projective Residual Update (PRU), as introduced in paper [18], aims to effectively remove specific data points from trained machine learning models. Initially designed for linear models such as logistic and linear regression, PRU's methodology extends to nonlinear models by treating them as comprising a fixed feature mapping followed by a linear or logistic regression layer. This adaptation simplifies the update process by focusing on the linear components of the model's structure, particularly the final layers in deep neural networks.

PRU utilizes synthetic predictions to estimate how the model would predict the outputs for data points earmarked for removal using the current model parameters. These synthetic predictions are pivotal because they act as substitutes for the actual outputs that the model would produce if the identified data points were removed. In the context of linear regression models, for example, these predictions are straightforwardly computed as the dot product of the model's current weights with the feature vector of each data point x_i.

The primary objective of PRU is to adjust the model's current weights so that its predictions for these synthetic outputs closely align with the actual outputs of the removed data points. To achieve this alignment, PRU employs optimization techniques such as gradient descent.

Through iterative updates, the model's weights are adjusted based on the disparity between the synthetic predictions and the real outputs of the data points scheduled for removal.

Metric: PRU typically maintains low L2 distances, indicating minimal deviation from exact retraining, especially notable in scenarios with large deletion groups. PRU often shows superior performance in the backdoor injection attack metric compared to [10]. For instance, while [10] might achieve metrics averaging around 0.2, PRU could achieve significantly lower values like 0.05. A lower value in the backdoor injection attack metric indicates that PRU is effective in removing or mitigating the influence of injected features that could compromise privacy.

vii. Performance Unchanged Model Augmentation (PUMA)

Definition: PUMA [44] updates the model parameters θ to θ_{mod}, ensuring minimal disruption to the model's predictive capabilities post-data removal. PUMA's approach uses optimization principles, particularly leveraging the Hessian Vector Product for efficient parameter adjustments. Hessian Vector Product approximates the impact of changes in model parameters on the loss function gradients, crucial for optimizing θ in response to removed data points.

The technique involves two primary steps: First, PUMA formulates an optimization problem to derive the modified parameters from the original parameters and incorporating adjustments that mitigate the removal of D_f's influence. This step ensures that the model's overall performance criteria are preserved or improved. Second, PUMA optimizes the perturbation factors assigned to the remaining data points $Dr \backslash D_f$. These factors are optimized to minimize the performance degradation caused by the removal of D_f, balancing between sparsity and small changes using regularization techniques.

Metric: PUMA consistently outperforms traditional methods like Retrain Model and [29] in several key metrics evaluated in the paper. Specifically, it shows up to a 10% improvement over the original model's performance when assessing the ability to preserve model performance after gradually removing data points. Additionally, in terms of effectiveness in data removal, PUMA reduces the success rate of membership attacks by 20-30% compared to other techniques such as Amnesiac Machine Learning. Furthermore, PUMA demonstrates superior efficiency by executing operations 40-50% faster than competing approaches in scenarios involving random data removal.

vii. Unlearn Features and labels

Definition: Unlearning in [45] involves updating the model parameters when the dataset changes from the original dataset to a modified dataset. This update is achieved using influence functions, a concept from robust statistics that measure the impact of individual data points on the model's parameters. The technique calculates precise updates by using first-order and second-order derivatives to reflect the removal or correction of specific data points or features.

One significant aspect of this approach is its ability to handle feature revocation, which involves removing entire features from the model. The process starts by identifying data points where these features are non-zero, then constructing a modified version of the dataset where these features are set to zero. The model parameters are then adjusted to account for these changes. Despite the reduction in input dimensionality, the method ensures that the model's performance and integrity are maintained through appropriate adjustments derived from the model's linear transformations.

A key consideration in the practical implementation of this technique is its scalability to large and complex models, such as deep neural networks. Direct computation of the Hessian matrix for exact updates is computationally prohibitive in such cases. Therefore, the paper proposes an approximation method for the inverse Hessian matrix. This approach enables efficient secondorder updates that balance computational feasibility with maintaining the integrity of the model adjustments during unlearning.

Metric: Compared to traditional baselines like retraining and [29], the proposed technique achieves 28% improvement in speed while maintaining high fidelity in correcting unintended memorization and label poisoning. The technique corrects poisoned labels, particularly achieving 85% accuracy restoration with 2,500 poisoned labels.

Federated unlearning

a) FedEraser

Definition: FedEraser [46] introduces a federated unlearning methodology aimed at reducing the influence of specific client data on a global model within federated learning setups. The primary objective is to adjust the parameters of the global model w_{global} to mitigate the impact of individual client contributions without directly accessing or compromising client data privacy. This adjustment process involves iteratively modifying

w_{global} by subtracting a scaled version of the client's model parameters w_c, denoted as $\gamma \cdot w_c$, where γ controls the magnitude of adjustment.

To operationalize this unlearning process, FedEraser incorporates a client calibration ratio r, defined as $r = \frac{E_{cali}}{E_{loc}}$, where E_{cali} represents the loss of the global model after unlearning and E_{loc} signifies the loss of the client's local model. This ratio guides the extent to which the client's influence is adjusted in the global model, ensuring a balanced approach to privacy preservation and model performance.

Another critical parameter in FedEraser is the retaining interval Δt, which determines the frequency of updates to $\mathbf{w}_{global}$ during the unlearning process. By carefully selecting Δt, FedEraser maintains stability in the global model while iteratively reducing the influence of client-specific data contributions.

Metric: To compare FedEraser, the paper uses two baselines: FedRetrain, which involves retraining the global model from scratch without the target client's data, and FedAccum, which accumulates updates from multiple clients without specific unlearning. For the Adult dataset, the F1-score for MIAs on the original model is 0.714. After unlearning with FedEraser, the F1score drops to 0.563, compared to 0.571 for FedRetrain. The impact of the calibration ratio (r) is also assessed. For the Adult dataset, with r = 0.1, FedEraser achieves a prediction accuracy of 85.8% on target data in 10.1 seconds. With r = 1.0, accuracy decreases slightly by 0.5%, but time increases to 100.1 seconds.

b) FU with Knowledge disitllation

Definition To eliminate the contribution of a specific client N from the final global model M_F, the paper [47] proposes erasing all historical updates ΔM_i^t from this client for rounds $t \in [1, F-1]$. Given N clients participating in each round t, the global model update ΔM_t can be expressed as:

$$\Delta M_t = \frac{1}{N} \sum_{i=1}^{N} \Delta M_i^t \tag{26}$$

To remove the contribution ΔM_N^t of the target client N, the updated model ΔM_0^t is recalculated as:

$$\Delta M_0^t = \Delta M_t - \frac{1}{N} \Delta M_N^t \tag{27}$$

Summing up these updates across rounds gives the unlearning version of the final global model M_{0F}:

$$M_0^F = M_1 + \frac{N}{N-1}\sum_{t=1}^{F-1}\Delta M_0^t + \sum_{t=1}^{F-1}\varepsilon_t \tag{28}$$

where ε_t represents the necessary corrections (skew) due to the incremental learning property of FL. This process mitigates skew accumulation caused by earlier model updates. Knowledge distillation is employed to refine the unlearning model using the original global model M_F as a teacher and the skewed unlearning model as a student. Soft class prediction probabilities q_i are generated using a softmax function over logits z_i:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{29}$$

where T is a temperature parameter that controls the smoothness of the probability distribution. Higher values of T produce softer distributions, enhancing model generalization. These soft probabilities are utilized to label unlabeled data, effectively transferring knowledge from the original model. During distillation training, if labeled data is available, a weighted average approach is adopted using both hard labels (ground truth) and soft labels produced by the global model at high temperature T. This approach balances the objectives, giving higher weight to soft labels to improve robustness and generalizability. After distillation training, the temperature T is set to 1, refining the unlearning model M_0^F to produce discrete class probabilities suitable for testing scenarios.

Metric: This paper evaluates the proposed unlearning technique using standard metrics to assess its effectiveness in removing the target client's influence from the global model. Comparisons are made against a baseline method of retraining from scratch. Results indicate a reduction of the attack success rate to zero post-unlearning. Additionally, through knowledge distillation, the technique achieves model recovery with test accuracy closely matching that of retraining from scratch.

c) Efficient Realization

Definition :The technique detailed in the paper "The Right to be Forgotten in Federated Learning: An Efficient Realization with Rapid Retraining" [48] introduces a sophisticated approach to federated unlearning. Initially, the unlearning process begins with a federated data deletion operation. This results in locally deleted datasets, which contain subsets of

the original data with certain samples removed. The process continues with the application of rapid retraining techniques to update the global model in response to the changes in the local datasets. Unlike traditional retraining methods that require exhaustive updates to all model parameters, the proposed technique employs a selective parameter update strategy based on the Fisher Information Matrix (FIM). By utilizing the FIM, the unlearning process can efficiently compute second-order derivatives necessary for parameter updates while minimizing computational overhead. Furthermore, the technique incorporates momentum techniques to enhance the stability and convergence speed of the unlearning process. The technique has limitations regarding the accuracy of the FIM approximation and the potential for divergence in unstable FL environments. While momentum techniques help mitigate these issues, further research may be needed to address challenges related to approximation errors and model convergence.

Metric: In terms of evaluation metrics, the paper compares the proposed technique to baseline methods such as retraining from scratch. Key metrics include the speed-up factor, which measures the efficiency of the unlearning process, and the Symmetric Absolute Percentage Error (SAPE), which quantifies the difference in model performance between the proposed technique and baseline methods.

d) FedRecover

Definition FedRecover [49] is a method designed to recover a federated learning (FL) global model after it has been subjected to poisoning attacks.The first step in FedRecover is the storage of historical data. During each global round, the server stores the model updates submitted by each client. The second step is the detection of malicious clients. At some point, malicious clients are detected based on their submitted updates. Detection mechanisms are not part of FedRecover, but the method assumes that these clients can be identified and removed. The third step involves the estimation of true model updates. After detecting and removing the malicious clients, the server needs to estimate the true model updates that would have been contributed by non-malicious clients. This estimation process is based on the stored historical updates. The final step is the re-aggregation of estimated updates. Using the estimated true model updates, the server performs a re-aggregation process similar to the standard federated averaging (FedAvg) but excluding the contributions from detected malicious clients. This re-aggregation involves averaging the estimated updates to form a new global model, effectively recovering the model from poisoning attacks.

Metric: The performance of FedRecover was evaluated using Training Error Rate which measures the accuracy of the global model and Attack Success Rate (ASR), which assesses the effectiveness of the attack in altering the model's predictions. FedRecover was compared to the train-from-scratch method and fine-tuning using clean datasets. Results showed that FedRecover achieves Training Error Rate and Attack Success Rate nearly identical to trainfrom-scratch, even when False Negative Rate is up to 0.5. Specifically, the Training Error Rate curves for FedRecover almost overlap with those for train-from-scratch, except when False Negative Rate is large (e.g., False Negative Rate 0.4) for Federated Averaging. Fine-tuning required a large number of clean examples, around 1,000 examples, to achieve Training Error Rate and Attack Success Rate comparable to FedRecover.

V. Discussion of results

In this section, a discussion on machine unlearning techniques is presented, focusing on their need for fine-tuning and the level of unlearning they achieve. The techniques vary in their reliance on fine-tuning after modifying data to maintain model performance, and the analysis examines the prevalence of instance-level versus class-level unlearning strategies. Instance-level unlearning removes individual data points as requested, while class-level unlearning allows broader modifications by removing entire data clusters at once. Key insights and observations are also highlighted.

Data Based

The common thread among these techniques is the consequential need for subsequent adjustments to the model, either through fine-tuning or retraining, highlighted by Table VII. This necessity arises due to the disruption caused by modifying the training data distribution. When specific data points are altered or removed, the model's decision boundaries and learned representations may no longer align with the original data characteristics. Fine-tuning allows minor adjustments to recalibrate the model, while retraining involves more substantial updates to accommodate these changes effectively.

The predominance of data-based machine unlearning techniques targeting single or multiple class levels, as observed in Table VIII, can be attributed to the relative simplicity and efficiency of creating and managing patterns for unlearning entire classes. Creating

and maintaining patterns for unlearning entire classes is less resource-intensive than handling individual instances. For example, using a trojan trigger or mnemonic code for a whole class involves maintaining a single pattern per class, rather than a unique pattern for each data point. This significantly reduces the overhead in terms of storage and computational complexity, making it easier to implement and maintain. Consequently, the higher percentage of techniques focusing on class-level unlearning is a natural outcome of these efficiencies.

A shortcoming of these techniques, particularly those involving label changes, lies in their potential to inadvertently reveal sensitive information. Changing the labels of data points is a relatively straightforward method that can be implemented with minimal computational resources. By altering the labels, the technique effectively modifies the models training data, leading it to forget specific information. However, if the technique changes the label of every data point to the same new label, this uniformity could potentially expose patterns or anomalies in the data. Consistently redirecting data points to a single label might make it easier for an adversary to infer that these points were part of a removal process, thus compromising the intended privacy. To mitigate this risk, a more sophisticated approach might involve randomly selecting new labels or distributing the relabeled data points across multiple labels. This strategy would make it more difficult for an adversary to detect any specific patterns related to the unlearning process.

Table VII. Fine-tuning Requirements and Levels of Unlearning

Paper	Is fine-tuning necessary?	Notes
[26]	✓	Specific details such as the number of epochs or the amount of data required for retraining after applying the trojan trigger were not explicitly mentioned.
[27]	✓	Requires incremental retraining with a few epochs and a small amount of data after applying redaction.
[17]	✓	Requires a repair step involving fine-tuning on a subset of the original retain set to restore accuracy.
[28]	✓	Fine-tuning may suffice for minor adjustments without substantial model reconfiguration.

Single or multiple class level

Single instance level

Architecture Based

a) Modular Unlearning

Analyzing the various techniques reveals a common thread: the pursuit of minimizing retraining efforts, as shown in Table VIII. Each iteration in modular unlearning techniques builds upon previous advancements, striving to streamline the process of updating models after data removal. This collective endeavor tries to maintain model accuracy and efficiency within evolving data landscapes. Techniques like those employing differential privacy or encoded data representations exemplify this trend, aiming to reduce computational overhead and preserve model integrity without compromising on performance. The evolution towards more efficient retraining strategies underscores the field's maturation, reflecting ongoing efforts to operationalize machine unlearning in real-world applications. Furthermore, the predominance of instance-level unlearning techniques in modular architectures can be attributed to several factors rooted in their architectural design and operational requirements. This architectural granularity allows for targeted updates and adjustments at the level of individual instances within these partitions, as observed in Figure VII. This approach underscores a deliberate effort to refine model adaptations precisely where necessary, optimizing performance without overhauling entire datasets.

An observation from these advancements is the prevalence of SISA in the literature on machine unlearning. The majority of reviewed papers (21/31) reference SISA in their related work or comparative analyses, highlighting its foundational role. This widespread citation underscores SISA's influence as a benchmark for evaluating new methodologies and innovations in adaptive machine learning systems. By creating consistent methods for dealing with unlearning requests, SISA has driven progress in making models more adaptable and reducing privacy risks in changing data environments. Its enduring presence in scholarly discourse underscores its pivotal role in shaping the trajectory of modular unlearning research.

Table VIII. Fine-tuning Requirements and Levels of Unlearning

Paper	Is fine-tuning necessary?	Notes
[29]	✓	Details regarding the extent and methodology of fine-tuning required were not explicitly mentioned.

[30]	✓	Requires fine-tuning with a specific subset of data to address the adaptive nature of the unlearning process.
[16]	✓	Fine-tuning necessary to maintain performance consistency after implementing unlearning techniques.
[31]	✓	Incremental retraining required to ensure the integrity and accuracy of the model post unlearning.
[32]	✓	-
[33]	✓	-

Single or multiple class level

Single instance level

b) Gradient Ascent

Based on the analysis of various unlearning techniques, it becomes evident that these methods necessitate meticulous tracking of each training batch's contribution during model training. When a batch containing sensitive or unwanted data is identified for removal, the unlearning process typically involves subtracting the accumulated parameter updates associated with those data points from the final model parameters. However, this efficiency comes with its own set of challenges and trade-offs. Storing the indices of examples participating in each batch and their corresponding updates requires significant storage capacity. Additionally, this method might cause the model to be different from what it would have been if those updates were never made, especially with larger datasets and more complicated training processes.

Another observation from the literature of this type of technique is the predominant focus on forgetting at the instance level rather than at the level of entire classes, Table IX. There is an absence of methods explicitly designed to forget entire data classes. This is because these techniques unlearn one data point at a time, requiring meticulous tracking and recording of each training batch's contributions during the model training process. Consequently, as these techniques adjust model parameters based on specific instances, they experience a gradual loss of accuracy or performance with each new request for unlearning.

Given the existing research, retraining may not be immediately necessary when the unlearning involves straightforward adjustments to model parameters or for simpler models, Table IX. However, in more complex models such as deep neural networks, where parameters are highly interconnected and changes to individual data points can have ripple effects across the model, fine-tuning or retraining prove to be beneficial. This ensures that the model adapts to the new data distribution post-unlearning and maintains or improves performance on unseen data.

Table IX. Fine-tuning Requirements and Levels of Unlearning

Paper	Is fine-tuning necessary??	Notes
[34]	✓	Some retraining is usually performed afterward to restore model performance on non-target data.
[35]	x	-
[36]	✓	-
[19]	✓	It needs some epochs of training using clean data and the identified trigger patterns.

Single instance level

c) Teacher-Student

All the techniques involving the teacher-student framework inherently perform a form of fine-tuning due to their operational methodology, as evidenced in Table X. These techniques utilize an iterative process where the student model is progressively adjusted based on the guidance provided by the teacher model. This approach mirrors fine-tuning, where the student model undergoes incremental updates to align with the teacher's outputs and to unlearn specific data. The iterative nature of these adjustments ensures that the student model refines its performance continually, similar to how fine-tuning hones a pre-trained model for specific tasks. Consequently, the fine-tuning aspect is embedded in the core mechanism of these teacher-student techniques, making it a fundamental component of their operation.

An insight from examining these techniques is the balanced distribution at the scope of unlearning, showing an equal split between single/multiple class-level and single/multiple instance-level unlearning, Fig X. Techniques like bad teaching, where the student learns

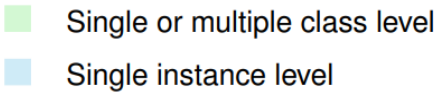
from both competent and incompetent teachers, enable class-level unlearning through generalized learning objectives. While other methods involving generative adversarial networks and gated knowledge transfer, that use pseudo samples to facilitate selective unlearning, ensure the student model forgets targeted information.

While reviewing the techniques described in the literature, it becomes evident that many rely heavily on distance functions to guide the process of unlearning. By leveraging distance functions, these methods aim to minimize the discrepancy between the original model and the adjusted model post unlearning. This ensures that retained knowledge remains intact while forgotten information is effectively erased or modified. However, the choice and design of these distance functions are pivotal, as they directly influence the effectiveness and efficiency of the unlearning process. inappropriate or overly complex distance metrics may introduce unnecessary computational overhead or obscure insights into model behavior.

The Student and Teacher Framework section reveals a prevalent trend towards iterative methodologies in machine unlearning techniques. The iterative nature ensures adaptability to dynamic datasets but also highlights a limitation: computational overhead due to repeated model adjustments. This iterative requirement suggests that while effectively managing targeted forgetting, these techniques may demand substantial computational resources, potentially limiting their scalability in real-time or resource-constrained environments. Upon reviewing the literature, it is also evident that another downside of these techniques is the reliance on maintaining more than one model simultaneously, which can be costly in terms of computational resources and storage.

Table X. Fine-tuning Requirements and Levels of Unlearning

Paper	**Is fine-tuning necessary??**
[37]	✓
[15]	✓
[38]	✓
[39]	✓
[41]	✓
[50]	✓
[42]	✓



d) Scrubbing Weights Approach

The Scrubbing Weights Approach encompasses a diverse array of techniques designed to effectively eliminate the influence of specific data points from machine learning models. These methods leverage sophisticated mathematical frameworks such as Hessians and Fisher Information Matrices to ensure precise data removal grounded in theory. Incorporating controlled noise into model weights is also a commonly adopted strategy that ensures efficient forgetting without compromising model performance. To tackle computational complexities in handling large and intricate models, some approaches utilize influence functions and approximate Hessian matrices, ensuring scalability and practical feasibility.

However, while these techniques offer robust solutions, they also present inherent limitations. Despite avoiding full retraining, they often require intricate mathematical computations and approximations, such as calculating Fisher Information Matrices or approximating Hessians, which can introduce computational overhead. Some methods rely on linear approximations or synthetic predictions, which may not fully capture the complexities of nonlinear models, potentially leading to suboptimal forgetting outcomes. Successful implementation hinges on accurate parameter estimation and transformation, with errors in these approximations posing risks to the effectiveness of data removal. Moreover, while efforts are made to minimize residual information, challenges persist in ensuring complete data erasure and preventing information leakage.

A strength identified in the literature is that most of these techniques do not necessitate retraining or fine-tuning, Table XI. Instead, they directly adjust model parameters based on calculated modifications that counteract the influence of targeted data points. This characteristic allows these methods to operate as single-step post-processing procedures, significantly reducing computational time and resource requirements compared to iterative retraining approaches. One method [18] deviates by iteratively adjusting model weights, resembling retraining processes to a certain extent, unlike the straightforward approach of other techniques.

Table XI. Fine-tuning Requirements and Levels of Unlearning

Paper	**Is fine-tuning necessary??**

[9]	x
[40]	x
[43]	x
[10]	x
[18]	✓
[44]	x
[45]	x

Single or multiple class level

Single instance level

Federated unlearning

In the realm of federated unlearning techniques, the necessity for finetuning or retraining varies based on the approach taken by each method. Generally, these techniques aim to mitigate the impact of individual client contributions on the global model without resorting to full retraining from scratch, aligning with the principle of efficiency in federated learning. Some techniques, such as FedEraser and Efficient Realization, integrate mechanisms that adjust the global model parameters iteratively by accounting for the influence of client-specific data without requiring fine-tuning. The specific fine-tuning requirements and levels of unlearning across these techniques are detailed in Table XII.

The approaches operate at the client level, where adjustments are made based on aggregated client updates rather than targeting specific data instances or entire classes. This distinction highlights that federated unlearning techniques do not uniformly align to either class-level or instance-level adjustments, emphasizing the need for methods that efficiently handle client contributions while maintaining global model performance and privacy.

Table XII. Fine-tuning Requirements and Levels of Unlearning

Paper	**Is fine-tuning necessary??**	**Notes**
[46]	x	-
[47]	✓	Fine-tuning process to integrate the distilled knowledge effectively
[48]	✓	Adjust the model parameters efficiently based on the FIM updates

[49]	x	-

Client level

Dataset

Analyzing dataset usage across different machine unlearning techniques provides insights into research trends and methodological choices within the field, as illustrated in Figure 4. CIFAR-10 emerges as a predominant choice across various techniques, reflecting its suitability for evaluating unlearning methods in complex image classification tasks. Its diverse range of objects and scenes allows researchers to assess model adaptability and robustness across different categories, ensuring a comprehensive evaluation of technique efficacy. MNIST, renowned for its simplicity and well-defined character recognition task, is frequently utilized in studies focusing on modular unlearning and scrubbing weights approaches. This dataset facilitates the evaluation of unlearning effects on basic classification tasks, providing insights into model behavior post-unlearning. A detailed table categorizing the datasets used in each referenced paper is presented in Appendix B for further reference.

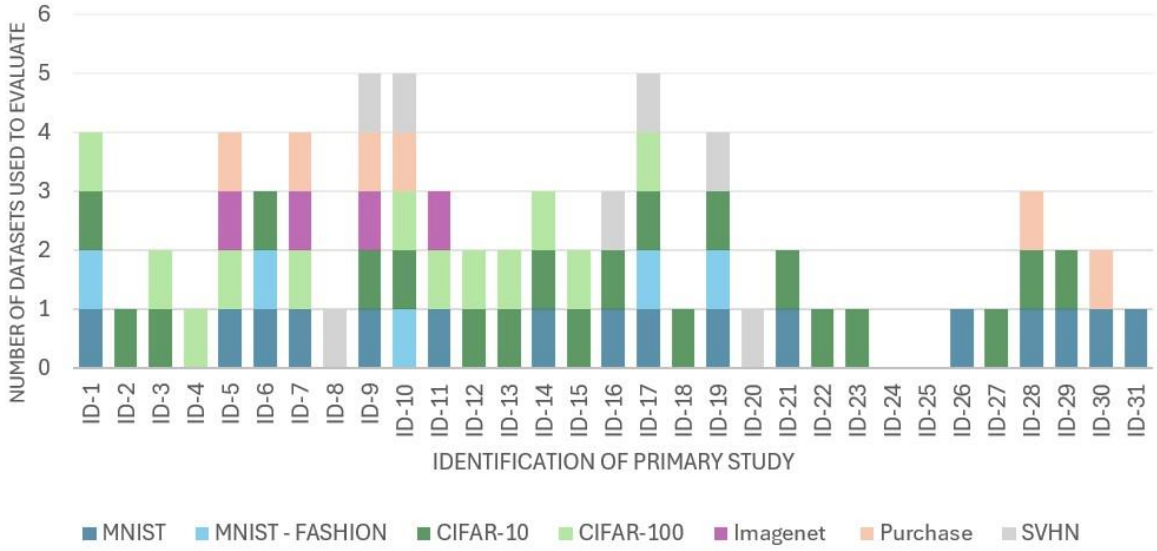
In contrast, specialized datasets such as HAM10000 and VGG-Faces feature prominently in teacher-student approaches, chosen for their relevance to specific applications like dermatology and facial recognition. These datasets enable researchers to evaluate unlearning techniques in contexts requiring nuanced model adjustments and fine-grained knowledge transfer between models. Imagenet, though less frequently used, appears in studies exploring modular unlearning and scrubbing weights approaches, leveraging its image diversity to assess technique performance in broader, more complex visual recognition tasks.

Papers like [10] and [18], which do not employ any datasets in their evaluations, indicate a focus on theoretical validation. This trend suggests a dual approach in machine unlearning research: while leveraging established benchmark datasets for generalizable insights, researchers also explore domain-specific datasets for task-specific evaluations. Moreover, almost all the datasets used across the analyzed papers are primarily intended for classification tasks. However, notable exceptions such as AgeDB, as observed in [42], demonstrate a unique suitability for regression tasks due to its annotation of age attributes for various subjects, encompassing a wide range of ages and identities. This dataset's utilization underscores the versatility of machine unlearning techniques beyond

classification, extending into domains requiring predictions of continuous outcome variables like age.

The analysis reveals a predominant use of convolutional neural networks in machine unlearning research. CNNs are used due to their effectiveness in image classification tasks, and ability to capture spatial hierarchies through convolutional layers. Despite the focus on CNNs, other neural network architectures, such as Recurrent Neural Networks and Long Short-Term Memory networks, could also be considered. These architectures are particularly useful for sequential data and time-series analysis, which opens possibilities for machine unlearning applications beyond image classification. Incorporating these neural network types could broaden the scope of machine unlearning research and provide insights into the unlearning process for different data modalities.

Fig. 4. Distribution of datasets utilized across different machine unlearning techniques



Architecture

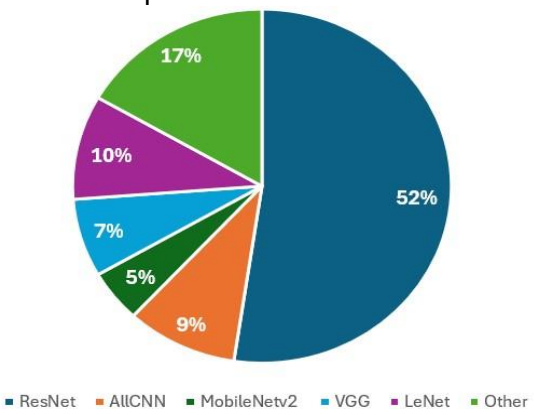
The analysis of architectures used in various techniques reveals several trends and preferences, as depicted in Fig 5 and in the more detailed table in Appendix C. ResNet, a family of convolutional neural networks, is extensively used across the studies. ResNet architectures, including ResNet-18, ResNet-50, and ResNet-20, are known for their residual learning framework which addresses the vanishing gradient problem by allowing gradients to flow through the network via shortcut connections [51]. The numbers in these architectures (18,50,20) refer to the depth of the network, specifically the number of layers. The increased depth in architectures like ResNet-50 allows for more complex feature extraction, while the residual connections help maintain the flow of gradients, thus

facilitating the training of very deep networks. Resnets robustness and versatility make it suitable for a wide range of machine unlearning evaluations. Another common architecture is the VGG family, particularly VGG-16, used in four papers. VGG-16 [52] employs a stack of convolutional layers with small receptive fields (3x3 filters) followed by fully connected layers. The number 16 in VGG-16 denotes the total number of layers in the network. VGG-16s design, with its consistent layer structure and uniform filter size, makes it computationally efficient and straightforward to implement. This simplicity is an advantage for benchmarking in machine unlearning research, as it allows for clear comparisons of model performance. The use of VGG-16 suggests a trend towards leveraging established architectures recognized for their performance in various computer vision tasks.

Including MobileNetv2 [53] in two studies indicates an interest in efficient and lightweight models. MobileNetv2 is designed for mobile and embedded vision applications, suggesting that researchers are considering the implications of deploying machine unlearning techniques in resource-constrained environments. Additionally, the use of DenseNet [54] underscores the importance of architectures that enhance feature reuse and reduce the number of parameters. DenseNets dense connectivity pattern helps mitigate the vanishing gradient problem and improves information flow, making it a choice for machine unlearning evaluations.

The table XV in Appendix C also highlights the diversity in architecture choices, with some studies employing multiple architectures to evaluate their techniques comprehensively. For example, one study [17] uses ResNet-18, All-CNN, and MobileNetv2, while another explores LeNet, ResNet, and VGG. This approach indicates a trend towards thorough benchmarking across different model complexities and capacities, ensuring that the proposed unlearning techniques are robust and generalizable. Futhermore the analysis also reveals that two papers do not specify any architectures, focusing instead on linear models and generalizing their findings to deep neural networks. These studies concentrate on theoretical validation, developing foundational principles that can be applied broadly. Despite generalizing their results to neural networks, these papers do not conduct specific experiments or evaluations involving particular neural network architectures.

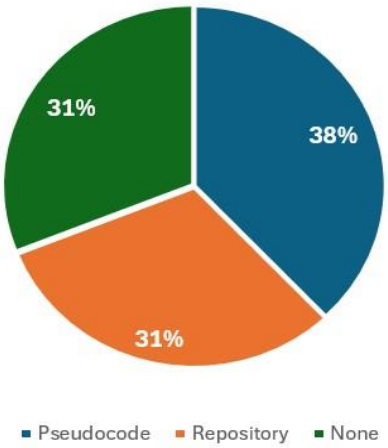
Fig. 5. Overview of Neural Network Architectures Utilized in Machine Unlearning Technique Evaluations



Replicability

The availability of source code or pseudocode significantly influences the replicability of machine unlearning techniques. Techniques, where authors provide comprehensive source code, facilitate easier replication by allowing other researchers to implement and verify the methods described directly. Pseudocode also plays a positive role in replicability. While it requires more interpretation than executable code, it provides a structured outline of the algorithmic steps involved. However, a notable portion of the techniques reviewed in the table either do not provide source code or pseudocode. This absence poses challenges to replication efforts, as researchers must rely solely on the methodological descriptions provided in the papers. Replicating these studies becomes more time-consuming and prone to interpretation errors, potentially leading to variations in results. As illustrated in Fig. 6, showing the distribution of papers with pseudocode, source code repositories, and those without. Further details can be found in Table XVI in Appendix D.

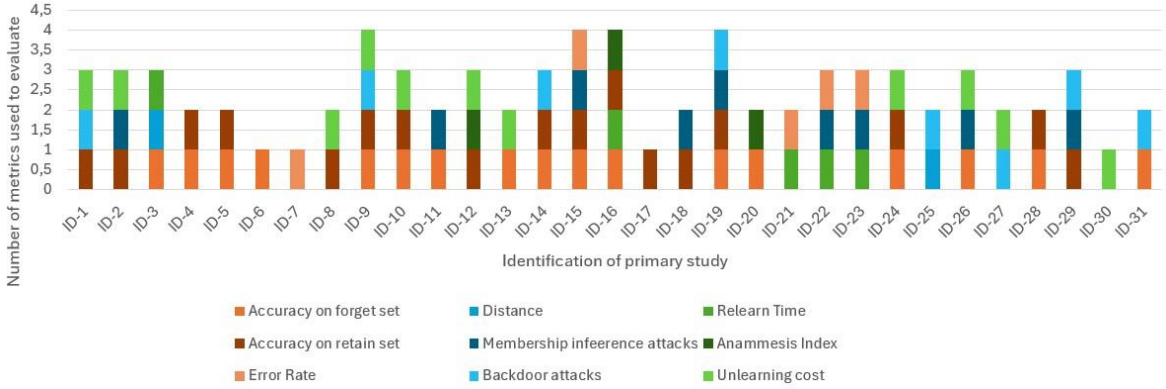
Fig. 6. Distribution of machine unlearning papers by the availability of code repositories, pseudocode, or neither.



Metrics

Fig.8 highlights a diverse range of metrics used across different studies to evaluate machine unlearning techniques. This variability suggests that no universal standard or consensus on which metrics are most appropriate for assessing the effectiveness of unlearning approaches. Many studies prioritize metrics related to privacy and security, such as membership inference and model inversion attack. These metrics are crucial for determining the extent to which unlearned models retain sensitive information and their vulnerability to privacy attacks. Metrics like accuracy on forgotten set and relearn time indicate studies' interest in understanding how unlearning techniques affect model performance. This consideration is essential for balancing privacy preservation with maintaining model effectiveness on retained data. Metrics such as unlearn time and activation distance reflect studies' concerns about the computational efficiency of unlearning techniques. Techniques that require less computational resources for unlearning are more practical for deployment in real-world applications.

Fig. 7. Distribution of metrics utilized across different machine unlearning techniques

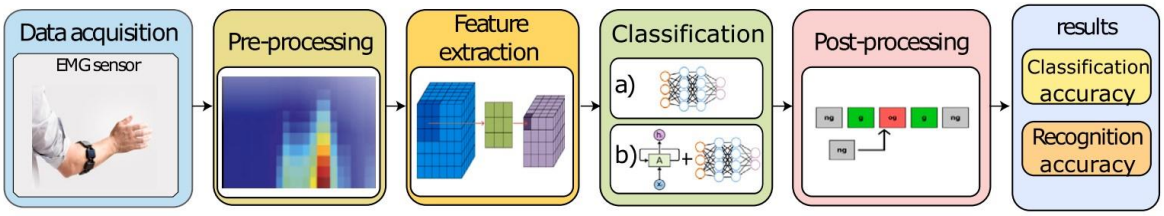


Case study: EMG signal classification

The recent paper [55] from the laboratory details a real-time hand gesture recognition system based on electromyographic (EMG) signals. The methodology involves data acquisition using EMG sensors, followed by preprocessing steps that include signal rectification and segmentation with a muscle activity detector. A sliding window approach is employed for feature extraction, resulting in vectors concatenated from multiple channels. The classification phase employs a CNN-LSTM model with 11,652,790 parameters, requiring approximately six hours to complete training. A significant challenge with EMG signals is their variability, not only between different users but also within the same person at different times. A potential future application of this classification system is in human-computer interfaces and prosthetics, where users can utilize EMG signals to perform

various activities on a computer. In this context, machine unlearning is particularly useful, not for privacy concerns, but to remove past data that is no longer relevant. Since EMG signals vary over time even for the same user, machine unlearning can help maintain the model's accuracy and relevance by removing outdated data, thus enhancing the system's adaptability and performance in real-time applications.

Fig. 8. Stages of a Hand Gesture Recognition model
Source: Adapted from [55]



When considering machine unlearning techniques for the EMG signal classification case, it is essential to evaluate various approaches and their applicability. This section explores the limitations of certain categories and highlights the potential of modular unlearning as a favorable solution. Techniques in the scrubbing weight category are not recommended for the EMG signal classification task. These methods are primarily theoretical, with many primary studies lacking repositories, pseudocode, or references to specific datasets. The absence of experimental validation and practical implementation renders these techniques unsuitable for real-world applications, especially where maintaining model performance in dynamic scenarios is critical. Data-based unlearning techniques, while offering a potential short-term solution, have notable limitations. These methods often fail to fully offset the influence of unlearned data in complex models. Moreover, they lack theoretical support for their validity, as evidenced by literature. Without guarantees on the extent of information retained by attackers or the similarity of parameters to a retrained model, data-based techniques do not provide the reliability needed for maintaining high accuracy and adaptability in EMG signal classification.

Modular unlearning emerges as a preferred option for the EMG signal classification task. Retraining the existing model, which takes approximately six hours for a model with 11,652,790 parameters, is feasible but can be significantly expedited with modular unlearning techniques. For instance, the DeepObliviate technique, which uses a ResNet-50 model with 25 million parameters, demonstrates a 10x speedup in the unlearning process. Unlike federated unlearning, which requires unlearning all contributions from client

models, modular unlearning allows for more granular control. Taking the SISA [29] technique as an example when an unlearning request arrives, the data point is removed from the relevant shards, and only the corresponding sub-models are retrained. This approach enables efficient data instance unlearning without the need to retrain the entire model. However, modular unlearning has its disadvantages. The process of sharding and managing multiple sub-models can introduce complexity and require careful coordination. Additionally, the need to retrain sub-models, while more efficient than full retraining, still incurs some computational overhead. Despite these challenges, modular unlearning represents a viable option for the EMG signal classification case. It offers a practical balance between implementation complexity and unlearning efficiency, making it a strong candidate for maintaining the accuracy and relevance of the model in real-time applications where EMG signals vary significantly over time.

VI. Future work and Conclusion

A. Emerging Challenges and Future Directions in Machine Unlearning Research.

As machine unlearning techniques become more prevalent, questions have emerged regarding their reliability and effectiveness in ensuring data privacy and model security. Researchers are increasingly seeking to understand the security implications and limitations of these methods, aiming to develop solutions that can effectively mitigate privacy risks. Parallelly, there is skepticism about the extent to which these techniques truly "forget" data points, as the veracity of their claims comes under scrutiny. This skepticism underscores the need for thorough investigation into the foundational principles of machine unlearning, as well as the development of practical frameworks for assessing their efficacy. In light of these challenges, this section delves into the emerging complexities and future directions in machine unlearning research, emphasizing the importance of addressing these issues to ensure the continued advancement and responsible deployment of machine unlearning technologies.

Unlike traditional membership inference attacks, the approach in [56] leverages outputs from both the original and unlearned models, utilizing various aggregation methods to combine the two posteriors for attack model input. The adversary’s objective is to determine if a target sample was unlearned from the original model, thus revealing potential privacy risks. Furthermore, the evaluation of the retraining from scratch method indicates that the attack degrades the membership privacy of the model. This highlights the unintended privacy risks posed by machine unlearning techniques.

The attack on machine unlearning techniques outlined in the paper [57] introduces a novel approach to exploiting vulnerabilities in the process of unlearning, particularly in scenarios where users actively manipulate their data to induce unlearning. The attack targets machine unlearning technique [10]. By strategically poisoning a subset of instances in the training data and submitting erasure requests, the attacker aims to slow down the unlearning process, ultimately diminishing or nullifying the efficiency gains of unlearning over full retraining. This attack highlights a critical vulnerability in machine unlearning systems, raising questions about the trade-offs between computational cost, model accuracy, and privacy in adversarial learning environments. Furthermore, the study emphasizes the need for robust defenses against such attacks and suggests avenues for future research to explore mitigating strategies.

The caution against solely comparing parameters to assess machine unlearning techniques stems from the paper [58]'s observation that datasets containing the points to be unlearned can generate similar parameter spaces as datasets where those points are forgotten. The paper reveals that during both learning and unlearning, these algorithms may traverse similar trajectories in the parameter space, leading to convergence on similar parameter configurations. This convergence can occur even when the datasets used in the learning and unlearning processes differ significantly. Subsequently, the authors highlight that the crucial factor determining the effectiveness of unlearning is not solely the resulting parameters but rather the assessing framework must give greater importance to the analysis of the process and algorithm. Consequently, machine unlearning techniques ([9], [40], [34]) that rely solely on parameter space comparisons to assess unlearning efficacy are susceptible to inaccuracies and may fail to provide robust guarantees of data removal or forgetting.

[59] presents a framework designed for the quantitative evaluation of compliance verification within the domain of machine unlearning. This paper hypothesis uses a testing techniques for assessing the efficacy of data deletion requests. The study introduces the concept of "privacy enthusiasts," individuals who actively participate in the verification process by embedding distinct backdoor patterns into their data prior to submission to a machine learning service. By measuring the success rate of these backdoors before and after submitting a deletion request, they can infer if their data was deleted. Results show that the approach can distinguish between compliant and non-compliant servers, even

when adaptive defenses are employed. Additionally, the research demonstrates the method's versatility across various machine learning systems and datasets, highlighting its potential applicability in diverse real-world scenarios.

Furthermore, this section explores approaches aimed at enhancing established machine unlearning (MU) methods. By exploring techniques like "prune first, then unlearn" and "sparsity-aware unlearning," the paper [22] delves into advancements that hold promise for improving the efficacy and efficiency of existing MU methods. One specific technique within magnitude-based pruning is one-shot magnitude pruning (OMP), which directly prunes the model weights to the target sparsity ratio based on their magnitudes in a single iteration. By integrating model sparsity into the unlearning process, methods like Fine-tuning, Gradient Ascent, Fisher Forgetting, and Influence Unlearning demonstrate improvements in unlearning accuracy and membership inference attack efficacy, without significant loss in remaining accuracy. Additionally, the paper integrates an ℓ_1 norm-based sparse penalty into the unlearning objective function, promoting model sparsity during the unlearning process. By incorporating this sparse penalty, the unlearning process prioritizes the retention of important model weights while reducing the magnitudes of 'unimportant' weights. Sparsity-aware unlearning demonstrates improvements in UA and MIA-Efficacy, effectively closing the performance gap with the gold-standard retrained-from-scratch model.

B. Conclusion

In this study, a taxonomy of machine unlearning techniques was developed, categorizing each approach based on its operational principles and mathematical underpinnings where applicable. The analysis and summarization of these techniques illuminate their diverse applications across various domains. From federated learning methods to data-based approaches, the study highlights the adaptability and nuanced strategies required for effectively managing outdated or sensitive data within machine learning models.

The evaluation metrics employed in assessing machine unlearning techniques demonstrated significant diversity, reflecting the multifaceted nature of post-unlearning performance evaluation. Metrics such as membership inference and model inversion attack underscored the critical need to assess privacy vulnerabilities, while accuracy on forgotten

sets and computational efficiency metrics provided insights into both model performance and operational feasibility.

In the context of specific applications like EMG signal classification, this study explored the effectiveness of modular unlearning techniques. Approaches such as DeepObliviate and SISA offer practical solutions for maintaining model accuracy while adapting to the dynamic nature of EMG signals over time. By selectively removing outdated data and minimizing computational overhead compared to full retraining, modular unlearning proves advantageous in scenarios requiring efficiency and real-time adaptability.

REFERENCES

[1] "Art. 17 GDPR - Right to erasure ('right to be forgotten') - GDPR.eu — gdpr.eu", https:// gdpr.eu/article-17-right-to-be-forgotten/, [Accessed 13-07-2024].

[2] "California Consumer Privacy Act (CCPA) — oag.ca.gov", https://oag.ca.gov/privacy/ccpa#heading5d, [Accessed 13-07-2024], 2024.

[3] C. Nast, "Now That Machines Can Learn, Can They Unlearn? — wired.com", https://www.wired.com/story/machines-can-learn-can-they-unlearn/, [Accessed 13-07-2024], 2021.

[4] T. Shaik, X. Tao, H. Xie, L. Li, X. Zhu, Q. Li, "Exploring the Landscape of Machine Unlearning: A Comprehensive Survey and Taxonomy", , 2024, URL: https://arxiv.org/abs/2305.
06360, 2305.06360.

[5] C. Li, "OpenAI's GPT-3 Language Model: A Technical Overview — lambdalabs.com", https:
//lambdalabs.com/blog/demystifying-gpt-3, [Accessed 13-07-2024], 2020.

[6] B. Kitchenham, "Procedures for performing systematic reviews", *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[7] F. Pedregosa, E. Triantafillou, Jun 2023, URL: https://blog.research.google/2023/06/announcing-first-machine-unlearning.html.

[8] Y. Cao, J. Yang, "Towards Making Systems Forget with Machine Unlearning", *in 2015 IEEE Symposium on Security and Privacy*, pp. 463–480, 2015, doi:10.1109/SP.2015.35.

[9] A. Golatkar, A. Achille, S. Soatto, "Eternal Sunshine of the Spotless Net: Selective Forgetting

in Deep Networks", , 2020, 1911.04933.

[10] C. Guo, T. Goldstein, A. Hannun, L. van der Maaten, "Certified Data Removal from Machine

Learning Models", , 2023, 1911.03030.

[11] A. Sekhari, J. Acharya, G. Kamath, A. T. Suresh, "Remember What You Want to Forget: Algorithms for Machine Unlearning", , 2021, 2103.03279.

[12] C. Dwork, A. Roth, "The algorithmic foundations of differential privacy", *Foundations and Trendso in Theoretical Computer Science˝* , vol. 9, pp. 211–407, 2013, doi:10.1561/0400000042.

[13] S. Bharati, M. R. H. Mondal, P. Podder, V. S. Prasath, "Federated learning: Applications, challenges and future directions", *International Journal of Hybrid Intelligent Systems*, vol. 18, no. 12, p. 1935, May 2022, doi:10.3233/his-220006, URL: http://dx.doi.org/10.3233/

HIS-220006.

[14] A. Menditto, M. Patriarca, B. Magnusson, "Understanding the meaning of accuracy, trueness and precision", *Accreditation and Quality Assurance*, vol. 12, no. 1, pp. 45–47, 2007, doi:

10.1007/s00769-006-0191-z, URL: https://doi.org/10.1007/s00769-006-0191-z.

[15] V. S. Chundawat, A. K. Tarun, M. Mandal, M. Kankanhalli, "Zero-Shot Machine Unlearning", *IEEE Transactions on Information Forensics and Security*, vol. 18, p. 23452354, 2023, doi:

10.1109/tifs.2023.3265506, URL: http://dx.doi.org/10.1109/TIFS.2023.3265506.

[16] K. Koch, M. Soll, "No Matter How You Slice It: Machine Unlearning with SISA Comes at the Expense of Minority Classes", , 2023.

[17] A. K. Tarun, V. S. Chundawat, M. Mandal, M. Kankanhalli, "Fast Yet Effective Machine Unlearning", *IEEE Transactions on Neural Networks and Learning Systems*, p. 110, 2024, doi:10.1109/tnnls.2023.3266233, URL: http://dx.doi.org/10.1109/TNNLS.2023.3266233.

[18] Z. Izzo, M. A. Smart, K. Chaudhuri, J. Zou, "Approximate Data Deletion from Machine Learning

Models", , 2021, URL: https://arxiv.org/abs/2002.10077, 2002.10077.

[19] Y. Liu, M. Fan, C. Chen, X. Liu, Z. Ma, L. Wang, J. Ma, "Backdoor Defense with Machine

Unlearning", , 2022, 2201.09538.

[20] T. Baumhauer, P. Schöttle, M. Zeppelzauer, “Machine Unlearning: Linear Filtration for Logit-

based Classifiers”, , 2020, 2002.02730.

[21] R. Chen, J. Yang, H. Xiong, J. Bai, T. Hu, J. Hao, Y. Feng, J. T. Zhou, J. Wu, Z. Liu, “Fast Model

Debias with Machine Unlearning”, , 2023, 2310.12560.

[22] J. Jia, J. Liu, P. Ram, Y. Yao, G. Liu, Y. Liu, P. Sharma, S. Liu, “Model Sparsity Can Simplify

Machine Unlearning”, , 2024, 2304.04934.

[23] M. Loog, T. Viering, “A Survey of Learning Curves with Bad Behavior: or How More Data Need Not Lead to Better Performance”, , 2022, URL: https://arxiv.org/abs/2211.14061,

2211.14061.

[24] S. C. J. B. MARTIN, “The Streisand Effect and Censorship Backfire”, , 2015.

[25] J. Serrà, D. Surís, M. Miron, A. Karatzoglou, “Overcoming catastrophic forgetting with hard attention to the task”, , 2018, URL: https://arxiv.org/abs/1801.01423, 1801.01423.

[26] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, X. Zhang, “Trojaning Attack on Neural Networks”, *in Network and Distributed System Security Symposium*, 2018, URL: https:// api.semanticscholar.org/CorpusID:31806516.

[27] D. L. Felps, A. D. Schwickerath, J. D. Williams, T. N. Vuong, A. Briggs, M. Hunt, E. Sakmar, D. D. Saranchak, T. Shumaker, “Class Clown: Data Redaction in Machine Unlearning at

Enterprise Scale”, , 2020, 2012.04699.

[28] T. Shibata, G. Irie, D. Ikami, Y. Mitsuzumi, “Learning with Selective Forgetting”, *in* Z.-H. Zhou, ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI21*, pp. 989–996, International Joint Conferences on Artificial Intelligence Organization, 8

2021, doi:10.24963/ijcai.2021/137, URL: https://doi.org/10.24963/ijcai.2021/137,

main Track.

[29] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie,

N. Papernot, “Machine Unlearning”, , 2020, 1912.03817.

[30] V. Gupta, C. Jung, S. Neel, A. Roth, S. Sharifi-Malvajerdi, C. Waites, “Adaptive Machine

Unlearning”, , 2021, 2106.04378.

[31] N. Aldaghri, H. Mahdavifar, A. Beirami, “Coded Machine Unlearning”, *IEEE Access*, vol. 9, p. 8813788150, 2021, doi:10.1109/access.2021.3090019, URL: http://dx.doi.org/10.1109/

ACCESS.2021.3090019.

[32] Y. He, G. Meng, K. Chen, J. He, X. Hu, “DeepObliviate: A Powerful Charm for Erasing Data

Residual Memory in Deep Neural Networks”, , 2021, 2105.06209.

[33] H. Yan, X. Li, Z. Guo, H. Li, F. Li, X. Lin, “ARCANE: An Efficient Architecture for Exact Machine Unlearning”, *in* L. D. Raedt, ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4006–4013, International Joint Conferences on Artificial Intelligence Organization, 7 2022, doi:10.24963/ijcai.2022/556, URL: https://doi.org/10.

24963/ijcai.2022/556, main Track.

[34] L. Graves, V. Nagisetty, V. Ganesh, “Amnesiac Machine Learning”, , 2020, 2010.10981.

[35] A. Thudi, G. Deza, V. Chandrasekaran, N. Papernot, “Unrolling SGD: Understanding Factors

Influencing Machine Unlearning”, , 2022, 2109.13398.

[36] Y. Liu, Z. Ma, X. Liu, J. Liu, Z. Jiang, J. Ma, P. Yu, K. Ren, “Learn to Forget: Machine Unlearning

via Neuron Masking”, , 2021, 2003.10933.

[37] V. S. Chundawat, A. K. Tarun, M. Mandal, M. Kankanhalli, “Can Bad Teaching Induce Forgetting? Unlearning in Deep Networks using an Incompetent Teacher”, , 2023, 2205.

08096.

[38] J. Kim, S. S. Woo, “Efficient Two-stage Model Retraining for Machine Unlearning”, *in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4360–4368, 2022, doi:10.1109/CVPRW56347.2022.00482.

[39] M. Kurmanji, P. Triantafillou, J. Hayes, E. Triantafillou, “Towards Unbounded Machine

Unlearning”, , 2023, 2302.09880.

[40] A. Golatkar, A. Achille, S. Soatto, “Forgetting Outside the Box: Scrubbing Deep Networks of Information Accessible from Input-Output Observations”, , 2020, 2003.02960.

[41] K. Chen, Y. Wang, Y. Huang, “Lightweight machine unlearning in neural network”, , 2021,

2111.05528.

[42] A. K. Tarun, V. S. Chundawat, M. Mandal, M. Kankanhalli, “Deep Regression Unlearning”, ,

2023, 2210.08196.

[43] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, S. Soatto, “Mixed-Privacy Forgetting in

Deep Networks”, , 2021, 2012.13431.

[44] G. Wu, M. Hashemi, C. Srinivasa, “PUMA: Performance Unchanged Model Augmentation for

Training Data Removal”, , 2022, URL: https://arxiv.org/abs/2203.00846, 2203.00846.

[45] A. Warnecke, L. Pirch, C. Wressnegger, K. Rieck, “Machine Unlearning of Features and

Labels”, , 2023, URL: https://arxiv.org/abs/2108.11577, 2108.11577.

[46] G. Liu, X. Ma, Y. Yang, C. Wang, J. Liu, “FedEraser: Enabling Efficient Client-Level Data Removal from Federated Learning Models”, *in 2021 IEEE/ACM 29th International Symposium*

on Quality of Service (IWQOS), pp. 1–10, 2021, doi:10.1109/IWQOS52092.2021.9521274.

[47] C. Wu, S. Zhu, P. Mitra, “Federated Unlearning with Knowledge Distillation”, , 2022,

URL: https://arxiv.org/abs/2201.09441, 2201.09441.

[48] Y. Liu, L. Xu, X. Yuan, C. Wang, B. Li, “The Right to be Forgotten in Federated Learning: An

Efficient Realization with Rapid Retraining”, *in IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, IEEE, May 2022, doi:10.1109/infocom48880.2022.9796721, URL: http://dx.doi.org/10.1109/INFOCOM48880.2022.9796721.

[49] X. Cao, J. Jia, Z. Zhang, N. Z. Gong, “FedRecover: Recovering from Poisoning Attacks in Federated Learning using Historical Information”, , 2022, URL: https://arxiv.org/abs/

2210.10936, 2210.10936.

[50] K. Chen, Y. Huang, Y. Wang, “Machine unlearning via GAN”, , 2021, 2111.11869.

[51] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition”, , 2015,

URL: https://arxiv.org/abs/1512.03385, 1512.03385.

[52] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image

Recognition”, , 2015, URL: https://arxiv.org/abs/1409.1556, 1409.1556.

[53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, , 2019, URL: https://arxiv.org/abs/1801.04381, 1801.04381.

[54] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, “Densely Connected Convolutional

Networks”, , 2018, URL: https://arxiv.org/abs/1608.06993, 1608.06993.

[55] L. I. Barona López, F. M. Ferri, J. Zea, Ángel Leonardo Valdivieso Caraguay, M. E.

Benalcázar, “CNN-LSTM and post-processing for EMG-based hand gesture recognition”, *Intelligent Systems with Applications*, vol. 22, p. 200352, 2024, doi:https://doi.org/

10.1016/j.iswa.2024.200352, URL: https://www.sciencedirect.com/science/article/pii/S2667305324000280.

[56] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, Y. Zhang, “When Machine Unlearning Jeopardizes Privacy”, *in Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS 21, ACM, Nov. 2021, doi:10.1145/3460120.3484756, URL: http://dx.doi.org/10.1145/3460120.3484756.

[57] N. G. Marchant, B. I. P. Rubinstein, S. Alfeld, “Hard to Forget: Poisoning Attacks on Certified

Machine Unlearning”, , 2022, 2109.08266.

[58] A. Thudi, H. Jia, I. Shumailov, N. Papernot, “On the Necessity of Auditable Algorithmic Definitions for Machine Unlearning”, , 2022, 2110.11891.

[59] D. M. Sommer, L. Song, S. Wagh, P. Mittal, “Towards Probabilistic Verification of Machine

Unlearning”, , 2020, 2003.04247.

VII. Appendices

1) Appendix A: Assignment of identifiers for primary studies. Table XIII presents ID, name of the study and the name of the technique proposed.

Table XIII. Identifier, Title of the Paper, Reference, and Name of the Technique

ID	Title of the Paper	Name of the technique
ID-1	Trojaning Attack on Neural Network	BadNets
ID-2	Class Clown: Data Redaction in Machine Unlearning at Enterprise Scale	Class Clown
ID-3	Fast Yet Effective Machine Unlearning	Fast yet effective machine unlearning
ID-4	Learning with Selective Forgetting	Mnemonic code
ID-5	Machine Unlearning	SISA
ID-6	Adaptive Machine Unlearning	Adaptive Machine Unlearning
ID-7	No Matter How You Slice It: Machine Unlearning with SISA Comes at the Expense of Minority Classes	No matter how you slice it
ID-8	Coded Machine Unlearning	Coded machine unlearning
ID-9	DeepObliviate: A Powerful Charm for Erasing Data Residual Memory in Deep Neural Network	DeepObliviate
ID-10	ARCANE: An Efficient Architecture for Exact Machine Unlearnine	ARCANE
ID-11	Amnesiac Machine Learning	Amnesiac Machine unlearning
ID-12	Unrolling SGD: Understanding Factors Influencing Machine Unlearnine	Unrolling SGD
ID-13	Backdoor Defense with Machine Unlearnin	BAERASER
ID-14	Learn to Forget: Machine Unlearning via Neuron Masking	Forsaken
ID-15	Can Bad Teaching Induce Forgetting?	Bad teaching
ID-16	Zero-Shot Machine Unlearning	Gated Knowledge Transfer
ID-17	Efficient Two-stage Model Retraining for Machine Unlearning	Efficient two-stage model
ID-18	Towards Unbounded Machine Unlearning	Towards Unbounded Machine Unlearning
ID-19	Lightweight machine unlearning in neural network	Lightweight machine unlearning
ID-20	Deep Regression Unlearning	Deep Regression Unlearning
ID-21	Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks	Eternal Sunshine

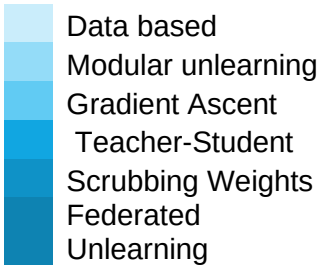
ID-22	Forgetting Outside the Box: Scrubbing Deep Networks of Information Accessible from Input-Output Observation	Forgetting outside the box
ID-23	Mixed-Privacy Forgetting in Deep Networks	Mixed privacy
ID-24	Certified Data Removal from Machine Learning Models	Certified Removal
ID-25	Approximate Data Deletion from Machine Learning Model	Projective Residual Update
ID-26	PUMA:Performance Unchanged Model Augmentation for Training Data Removal	Performance Unchanged Model Augmentation
ID-27	Machine Unlearning of Features and Labels	Unlearn Features and labels
ID-28	FedEraser: Enabling Efficient Client-Level Data Removal from Federated Learning Model	FedEraser
ID-29	Federated Unlearning with Knowledge Distillation	FU with Knowledge disitllation
ID-30	The Right to be Forgotten in Federated Learning: An Efficient Realization with Rapid Retraining	Efficient Realization with Rapid Retraining
ID-31	FedRecover: Recovering from Poisoning Attacks in Federated Learning using Historical Information	FedRecover

2) Appendix B: Comparison of the dataset used in each technique: Table XIV presents a comprehensive overview of the datasets utilized in the experimental evaluations conducted in each referenced paper. The table categorizes the datasets based on the specific approaches employed. Each row within the table corresponds to a distinct paper, detailing the datasets applied in the experimentation process. Also, each row is color-coded to indicate the technique's category within the taxonomy of machine unlearning approaches.

Table XIV. Datasets Used in Experimentation.

ID	MNIST	MNIST - FASHION	CIFAR -10	CIFAR -100	ImageNet	Purchase	SVHN	OTHER
ID-1	1	1	1	1	0	0	0	0
ID-2	0	0	1	0	0	0	0	0
ID-3	0	0	1	1	0	0	0	0
ID-4	0	0	0	1	0	0	0	Stanford Cars
ID-5	1	0	0	1	1	1	0	0
ID-6	1	1	1	0	0	0	0	0
ID-7	1	0	0	1	1	1	0	0
ID-8	0	0	0	0	0	0	0	Computer Activity dataset
ID-9	1	0	1	0	1	1	1	0
ID-10	0	1	1	1	0	1	1	0
ID-11	1	0	0	1	1	0	0	0
ID-12	0	0	1	1	0	0	0	0
ID-13	0	0	1	1	0	0	0	IMDB
ID-14	1	0	1	1	0	0	0	0
ID-15	0	0	1	1	0	0	0	Epileptic Seizure Recognition
ID-16	1	0	1	0	0	0	1	0
ID-17	1	1	1	1	0	0	1	HAM10000
ID-18	0	0	1	0	0	0	0	VGG-Faces
ID-19	1	1	1	0	0	0	1	0
ID-20	0	0	0	0	0	0	0	AgeDB, IMDB
ID-21	1	0	1	0	0	0	0	VGG-Faces
ID-22	0	0	1	0	0	0	0	VGG-Faces
ID-23	0	0	1	0	0	0	0	Caltech-256, MIT-67

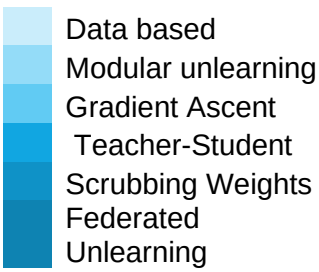
ID-24	0	0	0	0	0	0	0	0
ID-25	0	0	0	0	0	0	0	0
ID-26	1	0	0	0	0	0	0	Breast



3) Appendix C: Comparison of the architectures used in each technique: Table XV presents an overview of the architectures utilized in the experimental evaluations conducted in each referenced paper.

Table XV. Architectures Used in Various Studies

ID	Architecture
ID-1	ResNet-18
ID-2	CNN
ID-3	ResNet-18, All-CNN, MobileNetv2
ID-4	ResNet-18
ID-5	Wide ResNet-1-1, ResNet-50
ID-6	convolutional neural network
ID-7	ResNet- 18
ID-8	3-hidden-layers MLP
ID-9	LeNet, ResNet, VGG
ID-10	LeNet , Wide ResNet, ResNet-18
ID-11	Resnet-18
ID-12	VGG-16
ID-13	VGG-16
ID-14	ResNet-18
ID-15	ResNet-18, ResNet-34, MobileNetv2
ID-16	All-CNN, LeNet , ResNet-9
ID-17	ResNet-50
ID-18	All-CNN, ResNet-18
ID-19	Lenet, ResNET-18,VGG16,wide resnet
ID-20	ResNet-18
ID-21	ResNet-18
ID-22	All-CNN, Resnet-18
ID-23	ResNet50
ID-24	-
ID-25	-
ID-26	DenseNet
ID-27	CNN
ID-28	CNN
ID-29	VGG11, AlexNet
ID-30	ResNet-18, AlexNet
ID-31	ResNet-20



4) Appendix D: Availability of Source Code or Pseudocode: Table XVI provides an overview of the source code or pseudocode availability for the machine unlearning techniques discussed in the reviewed papers.

Table XVI. Source Code or Pseudocode Availability

ID	Do the primary studies include Pseudocode, a repository, or neither?
ID-1	Pseudocode
ID-2	-
ID-3	https://github.com/vikram2000b/Fast-Machine-Unlearning
ID-4	-
ID-5	https://github.com/cleverhans-lab/machine-unlearning
ID-6	Pseudocode
ID-7	-
ID-8	-
ID-9	Pseudocode
ID-10	-
ID-11	-
ID-12	https://github.com/cleverhans-lab/unrolling-sgd
ID-13	Pseudocode
ID-14	Pseudocode
ID-15	https://github.com/vikram2000b/bad-teaching-unlearning
ID-16	https://github.com/ayu987/zero-shot-unlearning
ID-17	Pseudocode
ID-18	https://github.com/Meghdad92/SCRUB
ID-19	Pseudocode
ID-20	https://github.com/ayu987/deep-regression-unlearning
ID-21	-
ID-22	-
ID-23	-
ID-24	-
ID-25	-
ID-26	Pseudocode
ID-27	Pseudocode
ID-28	https://www.dropbox.com/s/1lhx962axovbbom/FedEraser-Code.zip?dl=0
ID-29	Pseudocode
ID-30	github.com/yiliucs/federated-unlearning
ID-31	Pseudocode