A Survey on Federated Unlearning: Challenges, Methods, and Future Directions

ZIYAO LIU, Nanyang Technological University, Singapore, Singapore
YU JIANG, Nanyang Technological University, Singapore, Singapore
JIYUAN SHEN, Nanyang Technological University, Singapore, Singapore
MINYI PENG, Nanyang Technological University, Singapore, Singapore
KWOK-YAN LAM, Nanyang Technological University, Singapore, Singapore
XINGLIANG YUAN, The University of Melbourne, Melbourne, Australia
XIAONING LIU, RMIT University, Melbourne, Australia

In recent years, the notion of "the right to be forgotten" (RTBF) has become a crucial aspect of data privacy for digital trust and AI safety, requiring the provision of mechanisms that support the removal of personal data of individuals upon their requests. Consequently, machine unlearning (MU) has gained considerable attention which allows an ML model to selectively eliminate identifiable information. Evolving from MU, federated unlearning (FU) has emerged to confront the challenge of data erasure within federated learning (FL) settings, which empowers the FL model to unlearn an FL client or identifiable information pertaining to the client. Nevertheless, the distinctive attributes of federated learning introduce specific challenges for FU techniques. These challenges necessitate a tailored design when developing FU algorithms. While various concepts and numerous federated unlearning schemes exist in this field, the unified workflow and tailored design of FU are not yet well understood. Therefore, this comprehensive survey delves into the techniques and methodologies in FU providing an overview of fundamental concepts and principles, evaluating existing federated unlearning algorithms, and reviewing optimizations tailored to federated learning. Additionally, it discusses practical applications and assesses their limitations. Finally, it outlines promising directions for future research.

CCS Concepts: • **Security and privacy → Human and societal aspects of security and privacy**; • **Computing methodologies → Distributed computing methodologies**; • **General and reference → Surveys and overviews;**

Additional Key Words and Phrases: Federated unlearning, digital trust, AI safety

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

Authors' Contact Information: Ziyao Liu, Nanyang Technological University, Singapore, Singapore; e-mail: liuziyao@ntu.edu.sg; Yu Jiang, Nanyang Technological University, Singapore, Singapore; e-mail: yu012@e.ntu.edu.sg; Jiyuan Shen, Nanyang Technological University, Singapore, Singapore; e-mail: jiyuan001@e.ntu.edu.sg; Minyi Peng, Nanyang Technological University, Singapore, Singapore; e-mail: minyi002@e.ntu.edu.sg; Kwok-Yan Lam, Nanyang Technological University, Singapore, Singapore, Singapore; e-mail: kwokyan.lam@ntu.edu.sg; Xingliang Yuan, The University of Melbourne, Melbourne, Victoria, Australia; e-mail: xingliang.yuan@unimelb.edu.au; Xiaoning Liu, RMIT University, Melbourne, Victoria, Australia; e-mail: xiaoning.liu@rmit.edu.au.

ACM 0360-0300/2024/10-ART2
https://doi.org/10.1145/3679014

ACM Reference Format:
Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. 2024. A Survey on Federated Unlearning: Challenges, Methods, and Future Directions. *ACM Comput. Surv.* 57, 1, Article 2 (October 2024), 38 pages. https://doi.org/10.1145/3679014

1 Introduction

With increasing concerns for personal data privacy protection, governments and legislators around the world have enacted rigorous data privacy regulations, such as GDPR [102], APPI [51] and CCPA [32]. Typically, as digital service providers capture personal data from their users or data owners for service development, such regulations require them to grant users the **right to be forgotten (RTBF)**, with the provision of mechanisms that allow them to request the removal of their personal data from digital records. Consequently, given the extensive adoption of data-intensive **machine learning (ML)** algorithms, RTBF enables users to purge their data, including the influence of these data, from both the training dataset and the trained ML model. This is where **machine unlearning (MU)** [6, 12, 43–46, 49, 78, 79, 95, 123, 136] steps in as a critical facilitator of this process, ensuring that personal data is effectively and responsibly removed, further strengthening data privacy and ethical data handling. As depicted in Figure 1, the primary objective of unlearning is to remove the impact of specific data points from a trained model, while preserving the overall performance of the model.

Building upon the core principles of MU and the concept of RTBF, **federated unlearning (FU)** [8, 20, 41, 54, 76, 100, 103, 113, 117, 122, 143] has emerged to confront the challenge of data erasure within the domain of **federated learning** (**FL**) settings [56, 92, 132, 139, 148]. In a typical FL system, multiple clients locally train their machine learning models, which are subsequently aggregated to construct a global model. Then the server distributes the updated global model to all clients for training in the subsequent FL round. These sequential steps continue to recur until the global model reaches convergence (see Section 3.2 for more details). As a result, the objective of FU is to enable the FL model to remove the impact of an FL client or identifiable information associated with a client's partial data, while maintaining the privacy guarantees of the decentralized learning process, as illustrated in Figure 2. A formal definition of FU is provided in Section 3.3.

However, in contrast to traditional machine unlearning, the unique characteristics of federated learning introduce new targets and challenges (see Section 2 for more details). Therefore, this survey delves into techniques, methodologies, and recent advancements in federated unlearning. We provide an overview of fundamental concepts and principles in FU design, evaluate existing FU algorithms, present a taxonomy, and review optimizations of FU tailored to federated learning settings.

Comparison with related surveys. Currently, there are some works that have been conducted to summarize machine unlearning [86, 95, 101, 106, 121, 135, 136]. However, few existing surveys perceive the construction of federated unlearning. In Reference [131], the concept of knowledge editing throughout the entire lifecycle of federated learning is explored. This survey categorizes relevant works based on the principles of exact learning and approximate learning. These categories, as described in previous machine unlearning taxonomies like those in References [135, 136], are not specifically designed for a federated setting and thus may not fully capture the unique characteristics inherent in FU designs. The survey conducted in Reference [114] focuses on an analysis of only privacy and security threats within FU systems, with extensive discussions on potential attacks and defensive measures. It pays particular attention to the issue of privacy leakage stemming from distinctions between the trained model and the unlearned

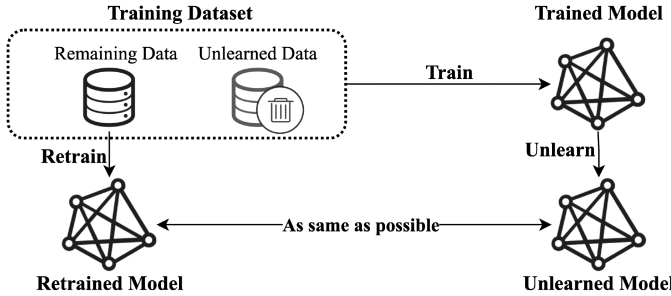

Fig. 1. Machine unlearning. Naive retraining, discarding the trained model and starting training from scratch with remaining data after unlearned data removal, is computationally intensive. Conversely, machine unlearning, which resumes training from the trained model through an unlearning process, is much more cost-effective. The objective of MU is to ensure that the unlearned model achieves a performance level on par with that of the retrained model.

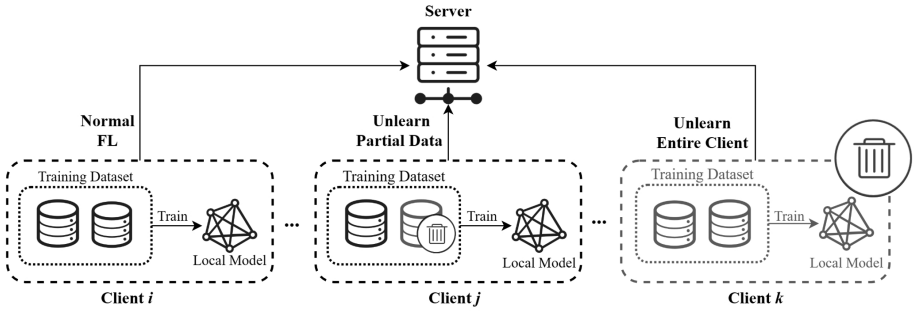

Fig. 2. Federated unlearning. In contrast to machine unlearning algorithms, which are typically executed by a single entity, FU systems involve multiple entities, including the unlearned client, remaining clients, and the central server, any of whom can act as the unlearner, responsible for executing the unlearning algorithm. Furthermore, the unlearning target may encompass either an entire client or specific partial data from a target client.

model, specifically examining their vulnerability to membership inference attacks. [138] provides a brief survey on federated unlearning, focusing on the level of data erasure, similar to the "unlearn-what" aspect discussed in our work. The works most closely related to ours are [103] and [53], which provide comprehensive surveys on existing federated unlearning literature. However, Reference [103] lacks an investigation into FL-tailored optimization and the limitations of existing approaches, while [53] lacks a formal definition of federated unlearning. Additionally, both [103] and [53] do not describe the unlearning workflow, which is important for readers to understand how unlearning integrates with **Machine Learning as a Service** (**MLaaS**). Furthermore, they do not specifically focus on security and privacy issues in federated unlearning systems.

While various concepts and numerous federated unlearning schemes exist in this field, the design and implementation of FU are still not fully explored. Furthermore, the methodology and principles for extending machine unlearning approaches to federated unlearning remain relatively unclear. The unified workflow of FU, particularly regarding security and privacy issues, is not yet well understood. This lack of comprehensive resources serves as the primary motivation for our effort in delivering this survey, which offers a deep and thorough insight into current FU research. A detailed comparison of related FU surveys is summarized in Table 1.

Table 1. Comparison of Related FU Surveys

Ref.	Def.		Taxonomy				Review				Insight			
	Target Formalization	Summary of Challenges	Unlearning Workflow	Who-unlearn	Unlearn-what	Who-verify	Comprehensive Review	Principle Analysis	Security and Privacy	Proof of Unlearning	FL-tailored optimization	Limitation	Experimental Evaluation	Future Directions
Wang et al. [114]	–	✓	–	✓	–	–	–	–	✓	–	–	–	–	–
Wu et al. [131]	–	✓	✓	–	–	–	–	✓	–	✓	✓	–	–	✓
Yang and Zhao [138]	✓	✓	–	–	✓	–	–	–	–	✓	✓	–	✓	✓
Nicolò et al. [103]	✓	✓	–	✓	✓	–	✓	✓	–	✓	–	–	✓	✓
Jeong et al. [53]	–	✓	–	✓	✓	–	✓	✓	–	✓	✓	✓	–	✓
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓

Summary of contributions. The main contributions of this survey are listed as follows.

(1) We present a unified federated unlearning workflow, on the basis of which we offer a novel taxonomy of existing FU techniques.
(2) Utilizing the proposed taxonomy and considering factors including (i) who-unlearn and (ii) unlearn-what, we conduct a comprehensive summary of existing federated unlearning methods, and highlight their distinctions, advantages, and constraints.
(3) We conduct a comprehensive examination of optimizations of FU techniques specifically tailored for federated learning, along with an assessment of their limitations.
(4) We delve deeply into critical discussions concerning the existing challenges in federated unlearning, and identify promising directions for future research.

Organization of the article. The rest of this article is organized as follows. Section 2 summarizes the targets, challenges, and characteristics of federated unlearning, and discusses their alignment. Section 3 describes the principles employed to achieve machine unlearning and provides an overview of the fundamentals of federated learning and unlearning. Section 4 presents different constructions of existing FU algorithms, followed by reviews of various optimizations tailored to federated learning and a critical examination of their limitations in Section 5. Section 6 offers discussions and outlines future research directions. Finally, Section 7 summarizes and concludes the article. An illustrative organization of the article is provided in Figure 3.

2 Targets and Challenges of Federated Unlearning

In this section, we will explore the targets of federated unlearning and the associated challenges compared with traditional machine unlearning. The insights gained will serve as a guideline for the taxonomy presented in Section 4.

2.1 Targets of Federated Unlearning

We now specify the targets of the unlearning process within an FL setting, for which the formal definitions are provided in Section 3.3.

Target 1 (Model Consistency). *The unlearned model must exhibit performance akin to a retrained model, ensuring the unlearning process neither diminishes its accuracy nor reliability.*

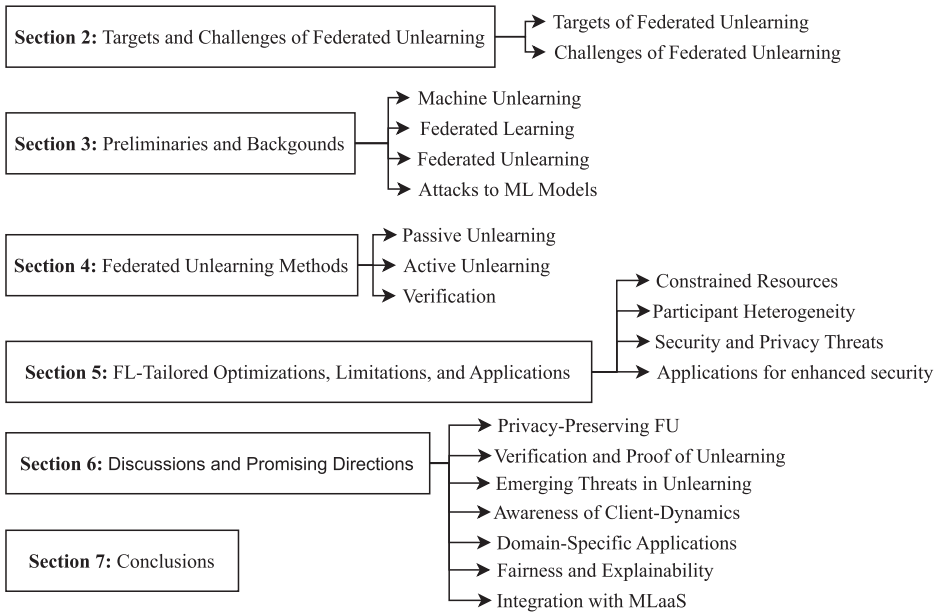

Fig. 3. Illustrative organization of the article.

Achieving this consistency is crucial, as it demonstrates the effectiveness of the unlearning algorithm in removing specific data while maintaining the model's overall quality.

Target 2 (Unlearning Efficiency). *As retraining in an FL system involves starting the training process from scratch, which is often inefficient, the target related to unlearning efficiency is to ensure that the cost of unlearning is significantly lower than that of obtaining a retrained model. These costs encompass various factors such as runtime, the number of participating clients, and communication overhead.*

Target 3 (Privacy Preservation). *FL is designed to offer privacy assurances by allowing access only to the locally trained model rather than the local data. Therefore, unlearning in a federated context must also ensure the preservation of clients' local data privacy. This approach ensures that while unlearning processes are implemented, the fundamental privacy guarantees of FL are maintained, safeguarding the privacy of clients' local data.*

Target 4 (Certified Removal). *The capability to verify the removal of either an entire FL client or partial data from a target client is essential. This process of certified removal should align with the unlearning request made by an FL participant. In other words, if the unlearning request is raised by an FL client, this client must be allowed to verify if its data has been unlearned and its impact on the FL model has been removed. Similarly, if the server raises the unlearning request, the server must also be able to monitor and verify the unlearning process. This verification process must be robust and reliable, ensuring that the removal adheres strictly to the specified unlearning request, thereby maintaining the trustworthiness of the federated unlearning system.*

2.2 Challenges of Federated Unlearning

In contrast to traditional machine unlearning, the unique characteristics of federated learning introduce certain challenges to the unlearning technique, as outlined below.

Challenge 1 (Knowledge Permeation). *When a client's data needs to be unlearned, its information has already spread throughout all participants in the FL system. This occurs because, during each FL round, the server aggregates the gradients from all clients and updates the global model. This updated model is then distributed to all clients, on which all clients conduct the subsequent round of FL training. As a result, the knowledge from the targeted client for unlearning permeates through to the other clients via the FL training process. Consequently, knowledge permeation complicates the achievement of model consistency (Target 1) in FU, compared to data-centralized MU schemes. Furthermore, implementing unlearning in a federated setting requires the involvement of all impacted clients, which significantly increases associated costs and impacts the target of unlearning efficiency (Target 2).*

Challenge 2 (Data Isolation). *Since every client individually maintains its dataset and conducts local model training, which is a key advantage of FL in terms of privacy preservation, only gradients or global models are publicly shared in an FL system. This aspect might hinder adapting existing MU algorithms, which rely on direct data access to be unlearned, within the FL context, aligning with the privacy preservation target (Target 3). Moreover, the absence of direct access to the unlearned data poses challenges in creating efficient FU algorithms, leading to concerns about unlearning efficiency (Target 2), compared to more efficient MU algorithms that directly utilize the unlearned data.*

Challenge 3 (Who-Unlearn). *Different from machine unlearning algorithms, which are typically executed by a single client, FU systems involve multiple participants, including (i) the unlearned client or target client,*[1] *(ii) remaining clients, and (iii) the central server, any of whom can act as the unlearner, responsible for executing the unlearning algorithm. Therefore, the FU algorithm selected by the unlearner depends on the degree of access to information about the data to be unlearned, consistent with the target of privacy preservation (Target 3). For example, when unlearning partial data of a target client, the target client possesses direct access to both the unlearned and remaining data, while the server's access is limited to historical data in the form of global models and gradients. Furthermore, when a client initiates a request for unlearning, it has the option to either participate in the unlearning process or simply exit the system. In cases where the target client chooses to leave, the unlearning process can be executed either on the server, the remaining clients, or both. Additionally, the entity responsible for unlearning also influences the need to verify that the unlearning process has been executed in accordance with the unlearner's request, contributing to the target of certified removal (Target 4).*

Challenge 4 (Unlearn-What). *In an FU system, the initiation of an unlearning request can stem from either the unlearned client or the server for different purposes. Concurrently, it's essential to consider that the unlearning target can be either (i) an entire target client or (ii) specific partial data from a target client. Considering the "unlearn-what" aspect, the unlearning principles of FU algorithms differ significantly. For example, when unlearning an entire target client, methods like local retraining, fine-tuning, and multi-task unlearning are no longer applicable (see Section 3.1.1 for more details). This is because these FU algorithms rely on direct access to the unlearned data and the remaining data of the target client, which becomes inaccessible when the entire client needs to be removed, in accordance with the target of privacy preservation removal (Target 3). The variation in designing FU algorithms for different unlearning targets also impacts the performance in achieving model consistency (Target 1), unlearning efficiency (Target 2), and certified removal (Target 4).*

Challenge 5 (Who-Verify). *As an FU system involves multiple participants, unlearning requests may be raised by FL clients or the FL server. For a more compatible scenario with the RTBF regulations, where unlearning services are provided by MLaaS infrastructures, clients must be allowed to verify*

[1]We use unlearned client and target client interchangeably.

Table 2. Alignment between Targets, Challenges and Characteristics of Federated Unlearning

Target	Challenge					Characteristic			
	Knowledge Permeation	Data Isolation	Who-Unlearn	Unlearn-What	Who-Verify	Constrained Resources	Participant Heterogeneity	Client Dynamics	Security and Privacy
Model Consistency	✓			✓				✓	
Unlearning Efficiency	✓	✓		✓	✓	✓	✓		
Privacy Preservation		✓	✓	✓				✓	✓
Certified Removal			✓	✓	✓			✓	

if their data has been unlearned and its impact on the FL model has been removed. Similarly, robust and provable proof of unlearning should be conducted on the server side if the request is raised by the server. However, implementing these verification processes presents significant challenges, including ensuring the efficiency (Target 2) and reliability (Target 4) of verification methods, maintaining system performance, and addressing potential security vulnerabilities. Only when data removal adheres strictly to the specified unlearning request can the trustworthiness of the federated unlearning system be maintained.

In addition to the primary distinctions and challenges highlighted above in FU in comparison to MU, several other factors may impede the effectiveness of federated unlearning. These factors arise from the unique characteristics of FL systems and are outlined as follows. Table 2 summarizes the alignment between these targets, challenges, and characteristics.

(1) **Constrained Resources:** In federated learning, devices or nodes that engage in the process often contend with constraints on their computing power, communication capabilities (such as limited network bandwidth), and storage capacities (like constrained memory). These limitations can affect their capacity to execute intricate model training tasks, facilitate efficient sharing and reception of updates, as well as manage the storage and processing of large ML models, datasets, or supplementary information. Consequently, resource-intensive MU algorithms may no longer be practical or scalable within the context of federated learning.
(2) **Participant Heterogeneity:** In FL systems, clients exhibit heterogeneity in various aspects, including their training capabilities related to factors such as data structure and distributions, e.g., vertical partitioned features and non-identically distributed data (Non-IID) data. This diversity necessitates the development of heterogeneity-aware FU approaches.
(3) **Client Dynamics:** In each FL round, clients are randomly chosen to participate in the model aggregation process. Besides, there may be a large number of dropped clients and newly-joined clients. The unlearner faces significant challenges in recalling past clients for unlearning operations, let alone retraining the model from scratch. These dynamic client behaviors can exert an influence on the effectiveness of machine unlearning algorithms, which were initially tailored for scenarios involving a single client in MU settings.

(4) **Security and Privacy Threats:** In FU settings, malicious attacks and information leakage are more intricate compared to a single-client MU scenario. Threat models become increasingly complex, taking into account factors like adversaries, their capabilities, and the potential for collusion.

3 Preliminaries and Backgrounds

In this section, we will first provide an overview of machine unlearning and summarize the principles of unlearning algorithms and metrics for the verification of unlearning. Then, we will provide an overview and formalization of federated learning and federated unlearning. Since attacks on ML models can be used for the verification of unlearning, an additional subsection is included to introduce attacks on ML models for completeness.

3.1 Machine Unlearning

In the MU system, the training dataset D consists of two components: D_u, representing the data samples to be forgotten, and D_r, representing the remaining data samples, where $D_r = D \backslash D_u$. We then consider $\mathcal{M}(D)$ as the final model trained on dataset D.

3.1.1 Unlearning Principles. Existing MU research papers predominantly rely on the following unlearning principles to make the distribution of the model $\mathcal{M}(D)$ identical to the distribution of the model $\mathcal{M}(D_r)$ [6].

Retraining. is a process training from a model free from the influence of data from D_u on the dataset D_r, essentially starting from scratch. In this method, a newly trained model $\mathcal{M}(D_r)$ does not have any information about D_u. However, this process is both time-consuming and resource-intensive because it discards the model $\mathcal{M}(D)$ on D containing the contribution of D_r dataset.

Fine-tuning. uses the remaining dataset D_r to optimize the model $\mathcal{M}(D)$ and reduce the impact of data from D_u. However, this process involves multiple iterations, leading to increased computational and communication costs.

Gradient ascent. represents a reverse learning process. In ML, the model $\mathcal{M}(D)$ is trained by minimizing the loss using gradient descent. Conversely, the unlearning process involves the application of gradient ascent to maximize the loss. However, this method can easily lead to catastrophic forgetting. As a result, many studies introduce constraints to preserve memory.

Multi-task unlearning seeks to not only eliminate the influence of D_u but also to reinforce the acquisition of knowledge from the remaining data D_r. In the course of these endeavors, most studies aim at striking a balance between the erasure effect and the retention effect.

Model scrubbing. applies a "scrubbing" transformation $\mathscr{H}$ to the model $\mathcal{M}(D)$ to ensure that the unlearned model closely approximates the perfectly retrained model with only D_r, as expressed by $\mathscr{H}(\mathcal{M}(D)) \approx \mathcal{M}(D_r)$ [30]. When defining the scrubbing method $\mathscr{H}$, most approaches rely on a quadratic approximation of the loss function. Specifically, for model parameters θ and ϕ, the gradient of the loss function of a given data point D_x satisfies

$$\nabla f_{D_x}(\phi) = \nabla f_{D_x}(\theta) + \mathcal{H}_{D_x}(\theta)(\phi - \theta),$$

where $\mathcal{H}_{D_x}(\theta)$ is positive semi-define. The scrubbed model becomes the new optimum by setting $\nabla f_{D_r}(\mathscr{H}_{D_r}(\theta)) = 0$, yielding the equation:

$$\mathscr{H}_{D_r}(\theta) = \theta - \mathcal{H}_{D_r}^{-1}(\theta)\nabla f_{D_r}(\theta).$$

$\mathscr{H}$ can perform a Newton step and can be derived under various theoretical assumptions [25] [31]. However, the challenge of this method lies in computing the Hessian matrix, which is infeasible for high-dimensional models. Therefore, some approaches aim at computing an approximation of the Hessian.

Synthetic data. is a method that replaces certain data with synthetic data to help the model "forget" specific information. An example of this approach involves generating synthetic labels for the data within D_u and then combining them with the data in D_u for training to accomplish unlearning. This method disentangles the impact of certain data from the model, helping to eliminate the influence of specific information while retaining the model's overall performance.

3.1.2 Verification. Verification methods aim at confirming whether data intended for deletion has indeed been effectively unlearned. Currently, these methods can be classified as outlined below:

Model performance. The most straightforward approach is to evaluate the model performance on the target client's data and test data to assess how effectively the data has been unlearned and how robustly the unlearned model is maintained. The evaluation metrics encompass accuracy, loss, and statistical errors.

Model discrepancy. Another approach to assess unlearning performance is by evaluating the discrepancy between the trained model and the unlearned model. This discrepancy can be measured using metrics such as Euclidean distance, KL-divergence, L2 distance, Wasserstein distance, and angle-based distance.

Execution efficiency. In addition, the time taken for the unlearning process, measured in terms of rounds, runtime, or speed-up ratio compared with a baseline, as well as memory consumption, can be used to evaluate the efficiency of the unlearning algorithm.

Attack performance. As introduced in Section 3.4, membership inference attacks can be used to determine whether a particular data was used during the training of a model. Therefore, by executing MIA on the unlearned model over unlearned data, the **attack success rate** (**ASR**) can be used to evaluate how effectively the data has been unlearned. Poorer performance by the MIA indicates that the influence of the unlearned data on the global model has diminished. Similarly, in the context of backdoor attacks, by injecting backdoors into the unlearned data and following the unlearning procedure, effective unlearning should disrupt the relationship between the trigger pattern and the backdoor class. The ASR can also be used to evaluate how effectively the backdoor is removed by unlearning. An empirical study on these metrics can be referred to in Reference [94].

3.2 Federated Learning

3.2.1 Overview of Federated Learning. The participants involved in federated learning [56, 92, 139] can be categorized into two categories: (i) a set of n clients denoted as $\mathcal{U} = u_1, u_2, \ldots, u_n$, where each client $u_i \in \mathcal{U}$ possesses its local dataset $\mathcal{D}_i$, and (ii) a central server represented as S. A typical FL scheme works by repeating the following steps until training is stopped [56]. (i) Local model training: each FL client u_i trains its model $\mathcal{M}_i$ using the local dataset $\mathcal{D}_i$. (ii) Model uploading: each FL client u_i uploads its locally trained model $\mathcal{M}_i$ to the central server S. (iii) Model aggregation: the central server S collects and aggregates clients' models to update the global model $\mathcal{M}$. (iv) Model updating: the central server S updates the global model $\mathcal{M}$ and distributes it to all FL clients.

3.2.2 Security and Privacy Threats in Federated Learning. The revelation of a participant's local model poses a direct threat to the fundamental privacy guarantee of standard federated learning [155]. Thus, privacy-preserving aggregation protocols [4, 5, 40, 84] are essential to maintain the security and privacy of the model aggregation process in Step iii of FL. Additionally, FL is susceptible to poisoning attacks [89] (see Section 3.4 for more details). In these attacks, malicious clients manipulate the global model by sending poisoned model updates to the server during Step ii, to affect global model performance or inject backdoors. Therefore, malicious-client detection mechanisms [7, 59, 109, 149] are imperative to differentiate between malicious and benign clients.

3.3 Federated Unlearning

In an FU system, the set of FL clients is represented as U, where each client $u_i \in U$ possesses a local dataset D_i. This set is categorized into two distinct subsets: U_u, which includes clients designated for unlearning (either entirely or partially), and U_r, comprising the remaining clients, with the relationship $U = U_r \cup U_u$. More specifically, for any client $u_j \in U_u$, $\bar{D}_j$ represents the data of u_j to be unlearned, hence $\bar{D}_j = D_j$ signifies unlearning of the entire client u_j, and $\bar{D}_j \subset D_j$ indicates unlearning partial data of the client u_j. Now, we give the definition of federated unlearning.

Definition 1 (Federated Unlearning). A federated unlearning process $FU(M, U, U_u, U_r) \rightarrow \bar{M}$ is defined as a function from a global model M obtained through FL $FL(U)$ trained by a set of FL clients U to an unlearned model $\bar{M}$. This function considers two subsets of U including the set of unlearned client $U_u \subset U$ where each $u_j \in U_u$ posses its unlearned dataset $\bar{D}_j$, and the set of remaining client $U_r \subset U$. The goal is to ensure that the unlearned global model $\bar{M}$ maintains performance comparable to a retrained model $\hat{M}$ trained by $U_r \cup U_u$ where each $u_i \in U_r$ posses D_i and each $u_j \in U_u$ posses $D_j \backslash \bar{D}_j$.

Building on the definition of federated unlearning outlined above, we specify the targets of the unlearning process within an FL setting as follows.

Definition 2 (Model Consistency). For a given set of samples X, let $\bar{Y}$ be the predicted results produced from the unlearned global FL model $\bar{M}$, and $\hat{Y}$ be the predicted results from a retrained global FL model $\hat{M}$. Then, the unlearning process $FU(M, U, U_u, U_r)$ is considered to provide full consistency if $\bar{Y} = \hat{Y}$. The target regarding model consistency is to make the performance of the unlearned model $\bar{M}$ as much as similar to that of $\hat{M}$.

Definition 3 (Unlearning Efficiency). For a retrained model $\hat{M}$ and an unlearned model $\bar{M}$ obtained from $FU(M, U, U_u, U_r)$ with full consistency, the target regarding unlearning efficiency is to make the cost of $FU(M, U, U_u, U_r)$ as much less than the cost of obtaining the retrained model $\hat{M}$.

Definition 4 (Privacy Preservation). For a federated learning process $FL(U)$ followed by a federated unlearning process $FU(M, U, U_u, U_r)$, the target regarding privacy preservation is to ensure that the additional information leakage caused by $FU(M, U, U_u, U_r)$, beyond what is leaked through $FL(U)$, is kept as minimal as possible.

Definition 5 (Certified Removal). For any participant, whether an FL client or server, initiating the unlearning request, the target regarding certified removal is to establish a function $V(\cdot)$, which serves to confirm that the unlearning process $FU(M, U, U_u, U_r)$ has been carried out in accordance with the request made by that participant.

3.4 Attacks to ML Models

As mentioned earlier, attacks on ML models can serve as a means to verify the effectiveness of unlearning. Specifically, these attacks can help determine whether the data related to the target client has been successfully unlearned. In this section, we will primarily introduce the two most widely adopted attack methods that are utilized for unlearning verification:

3.4.1 Membership Inference Attacks (MIA). First proposed by Shokri et al. [110], the fundamental idea behind MIA is to determine whether a particular record was used during the training of a target model. This is predicated on the observation that data samples present in the training set will lead the model to produce outputs with higher confidence scores. Consequently, an adversary can train a separate model for binary classification, designating outputs as either "member" (indicating that the data was part of the training set) or "non-member" (indicating that the data was not

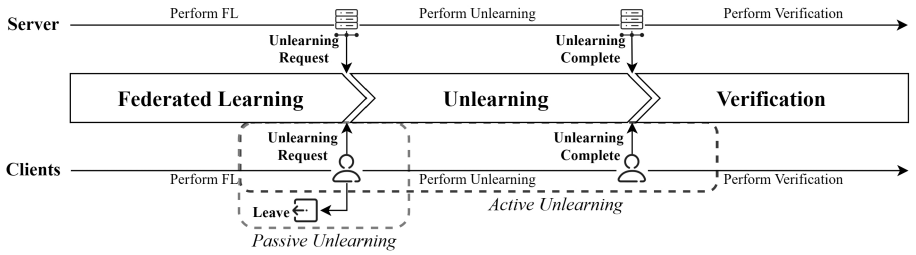

Fig. 4. An unified federated unlearning workflow. This workflow outlines the timeline for learning, unlearning, and verification. When the FU system receives an unlearning request, it can follow either the passive unlearning approach, where the target client exits the system immediately, or the "Active unlearning" approach, where the target client chooses to stay and participate in the unlearning process. Unlearning requests can be initiated by either the unlearned client or the server for various purposes. Furthermore, the unlearning and verification roles can be performed by the server, the target clients, the remaining clients, or a combination of both.

part of the training set). This potential to distinguish between member and non-member records poses a threat to data privacy. Remarkably, MIA does not require knowledge of the target model's specific architecture or the distribution of its training data. Relying on the shadow models, a series of shadow training datasets $D'_1, \ldots, D'_k$ and disjointed shadow test datasets $T'_1, \ldots, T'_k$ can be synthesized to mimic the behavior of the target model so as to train the attack model. Evaluating the unlearning effectiveness can be achieved by performing MIA on the unlearned model with the unlearned data. The ASR serves as an indicator of how well the data has been unlearned. A decrease in the MIA's performance suggests that the influence of the unlearned data on the global model has been successfully reduced.

3.4.2 Backdoor Attacks (BA). Backdoor attacks embed a distinct pattern or "trigger" into portions of the training data [65]. The trigger can be a small patch or sticker that is visible to humans [39, 73], or the value perturbation of benign samples indistinguishable from human inspection [2, 61, 104]. When the model is subsequently trained or fine-tuned on this, its behavior remains typical for standard inputs. Yet, upon detecting an input that contains this covert trigger, the model will yield malicious behaviors that align with the attacker's intentions. Backdoor attacks are particularly concerning because they can remain dormant and undetected until the attacker chooses to exploit them. In the context of unlearning, injecting backdoors into the unlearned data and then applying the unlearning procedure should effectively disrupt the relationship between the trigger pattern and the backdoor class. The ASR can be utilized to assess the effectiveness of the unlearning process in removing the backdoor. A lower ASR would indicate that the backdoor has been successfully eliminated.

4 Federated Unlearning Methods

In this section, we introduce a unified federated unlearning workflow, as illustrated in Figure 4, serving as the basis for a novel taxonomy of existing FU techniques. This workflow defines the timeline for learning, unlearning, and verification. When the FU system receives an unlearning request, it can either allow the target client to exit the system immediately, referred to as "Passive unlearning," or the target client can choose to stay and participate in the unlearning process, referred to as "Active unlearning." Note that some unlearned clients may simultaneously initiate the unlearning request and transmit information to the server, while others may not engage in the unlearning process but remain solely for verification. We categorize these FU schemes as passive

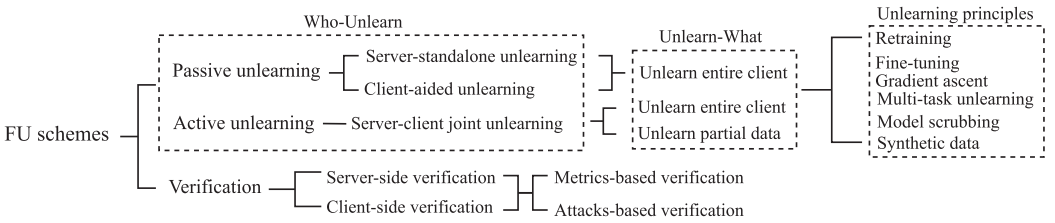

Fig. 5. Taxonomy of federated unlearning schemes.

unlearning as well. The taxonomy can be found in Figure 5 and the summary of the existing FU works can be found in Table 3.

4.1 Passive Unlearning

Passive unlearning signifies that the target client does not stay within the FU system to participate in the unlearning process, which typically involves a series of computational iterations. Instead, the remaining participants, including the central server, the remaining FL clients, or both, carry out the unlearning algorithms. In this case, passive unlearning unlearns the entire client instead of partial data. In the scenario of (i) server-standalone unlearning, historical information such as gradients and global models is stored, enabling the server to eliminate the influence of the unlearned client using various methods. In the scenario of (ii) client-aided unlearning, the standard FL workflow is followed, with iterative refinements of the global model achieved by aggregating improved information from the remaining clients. Note that for passive unlearning, methods like local retraining, local fine-tuning, and multi-task unlearning are no longer applicable. This is because these FU algorithms rely on direct access to both the unlearned data and the remaining data of the target client, which becomes inaccessible when the entire client must be removed.

4.1.1 Server-Standalone Unlearning. As previously mentioned, standalone server unlearning typically depends on the utilization of stored historical data, which may include gradients, global models [41, 49, 54, 129, 130, 146], contribution information [146], or intermediate information necessary for constructing a random forest [75]. This category necessitates a significant amount of memory on the server, potentially limiting its practical application in large-scale FL systems with complex ML models.

In the case of FedRecovery [146], the server retains historical data from all clients and quantifies their contributions in each round based on gradient residuals. When a target client requests to leave, the server systematically removes its contributions from all FL rounds through a fine-tuning process. Based on FedRecovery [146], Crab [54] achieves a more efficient recovery based on (i) selective historical information rather than all historical information and (ii) a historical model that has not been significantly affected by malicious clients rather than the initial model. Additional constraints can be introduced to further guide the recovery process, such as a penalty term based on projected gradients [26, 107], randomly initialized degradation models [152], and estimated skew [49]. The approach of eliminating the contribution of the target client is more straightforward in [129, 130], where the server directly averages the models of the remaining clients. Strategic retraining based on the change of sampling probability is adopted for fast and efficient recovery [113]. To mitigate the potential decrease in accuracy due to the averaging process in the averaged model, knowledge distillation is employed. This technique facilitates the transfer of information from the trained model to the unlearned model, helping to preserve performance. Consequently, these designs adhere to a multi-task unlearning approach. In VERIFI [28], after receiving the gradients from all clients, including those from the target client, are uploaded to

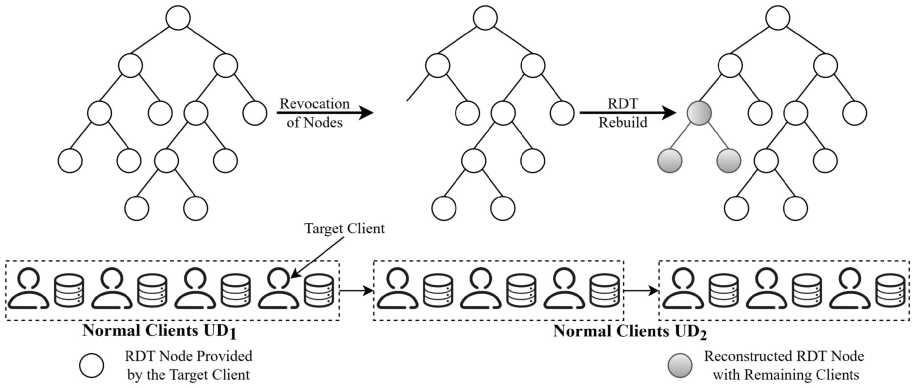

Fig. 6. An illustration of RevFRF [75]. To remove a target client, the server first identifies nodes affected by the target client and subsequently eliminates these affected nodes until reaching the leaf node. Following this, the server reconstructs the affected branches through a retraining process, based on previously stored intermediate information.

the server. The server then amplifies the gradients from the remaining clients and diminishes the gradients of the target client, to reduce the impact of target clients, hence achieving unlearning.

Apart from the above-mentioned works, RevFRF [75], as shown in Figure 6, focuses on federated random forest training. To remove a client, the server first identifies nodes affected by the target client and subsequently eliminates these affected nodes until reaching the leaf node. Subsequently, the server reconstructs the affected branches using previously stored intermediate information, rather than instructing the remaining clients to undergo retraining. In particular, in the worst-case scenario where the revoked node is the root node, the server has to reconstruct the entire random decision tree. Unlearning within a federated clustering setting is explored in SCMA [97], where each client maintains a vector to denote its local clustering result. These vectors are then aggregated by the server to form a global clustering outcome. Eliminating a client is straightforward by assigning a zero vector to the unlearned client and then re-aggregating all vectors.

Limitations. Server-standalone unlearning, which relies on historical data stored on the server, lacks real-time input from remaining clients during the unlearning process. This limitation may result in slightly lower unlearning performance compared with client-aided unlearning. This characteristic could impede the applicability of server-standalone unlearning in complex ML models or Non-IID FL settings, where there is a notable bias, affecting the overall efficacy of the unlearning process. Furthermore, server-standalone unlearning may lack responsiveness to changes in data and client behavior, as it solely relies on historical information. This can limit its adaptability in dynamic environments where real-time data and client interactions are crucial.

4.1.2 Client-Aided Unlearning. Unlearning performed by the server and remaining clients typically offers greater potential compared with standalone server unlearning. This is because the remaining clients contribute valuable information about the remaining data, which enables the server to enhance its unlearning process. In this context, the server may or may not have access to historical information.

This research direction is arguably pioneered by the design of FedEraser [71], as shown in Figure 7. The core concept of FedEraser is that the current global model can be reconstructed using only the initial model and historical clients' gradients at each round. Consequently, unlearning boils down to eliminating the influence of the target client on the historical gradients, i.e., calibrating historical gradients. To achieve this goal, for a historical FL round i with the stored gradients

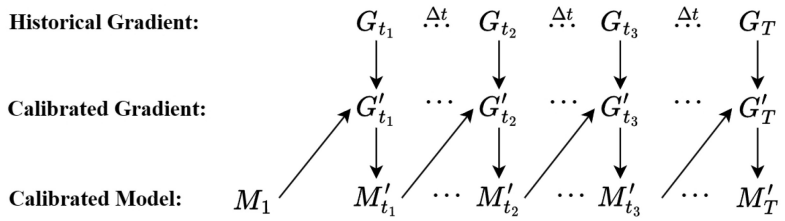

Fig. 7. An illustration of FedEraser [71]. The server stores clients' gradients at intervals of every Δt rounds. Using an iterative approach, for a given round t_i, the server computes calibrated gradients G'_{t_i} based on historical gradients G_{t_i} and the calibrated model $M'_{t_{i-1}}$.

G_i and a calibrated global model M_{i-1} for the previous round, each remaining client u_j calculates a local calibration direction c_i^j based on its local data D_j and M_{i-1}. The local calibration directions are then aggregated by the server to derive a global calibration direction c_i, which enables the server to calculate calibrated historical gradients G'_i and to obtain a calibrated global model M_i via G'_i. This iterative process continues round-by-round until all historical gradients are successfully calibrated, resulting in the server obtaining a final calibrated global model, eliminating the influence of the target client. To enhance unlearning efficiency, the server stores clients' gradients at intervals of every Δt rounds, leading to a tradeoff between unlearning performance and resource consumption in terms of memory and computation. Improve upon FedEraser, Crab [54] and Fast-FedUL [49] optimize storage efficiency by selectively storing important gradients, while Sharding Eraser [67] compress storage using coding-based techniques. A similar idea is adopted in [120] focusing on ranking tasks instead of classification tasks. Building upon the unlearning concept introduced in FedEraser, an efficiency-enhancing technique is employed in FRU [142] for federated recommendations. In FRU, only the important updates to clients' item embeddings are stored. In line with FedEraser [71] and FRU [142], FedRecover [8] also entails the storage of historical gradients and global models. In FedRecover, to prevent the remaining clients from computing exact model updates for fine-tuning, which can lead to significant computational overhead, the server calculates updates for the remaining clients using historical gradients and global models, as described below:

$$\boldsymbol{g}_t^i = \overline{\boldsymbol{g}}_t^i + \mathbf{H}_t^i \left(\hat{\boldsymbol{w}}_t - \overline{\boldsymbol{w}}_t\right)$$

where $\mathbf{H}_t^i = \int_0^1 \mathbf{H}\left(\overline{\boldsymbol{w}}_t + z\left(\hat{\boldsymbol{w}}_t - \overline{\boldsymbol{w}}_t\right)\right) dz$ is an integrated Hessian matrix for the ith client in the tth round. Denote the global-model difference in the tth round as $\Delta \boldsymbol{w}_t = \hat{\boldsymbol{w}}_t - \overline{\boldsymbol{w}}_t$ and the model-update difference of the ith client in the tth round as $\Delta \boldsymbol{g}_t^i = \boldsymbol{g}_t^i - \overline{\boldsymbol{g}}_t^i$. The Hessian matrix $\tilde{\boldsymbol{H}}_t^i$ for the ith client in the tth round can be approximated as

$$\tilde{\boldsymbol{H}}_t^i = \text{L-BFGS}\left(\Delta \boldsymbol{W}_t, \Delta G_t^i\right)$$

where $\Delta \boldsymbol{W}_t = \left[\Delta \boldsymbol{w}_{b_1}, \Delta \boldsymbol{w}_{b_2}, \ldots, \Delta \boldsymbol{w}_{b_s}\right]$ and $\Delta G_t^i = \left[\Delta \boldsymbol{g}_{b_1}^i, \Delta \boldsymbol{g}_{b_2}^i, \ldots, \Delta \boldsymbol{g}_{b_s}^i\right]$ are L-BFGS buffers [96] maintained by the server. Nonetheless, these approximations introduce estimation errors over rounds. Therefore, the remaining clients are periodically tasked with computing their exact model updates to correct these approximations, based on an adaptive abnormality threshold. A more straightforward retraining-based method is employed in SIFU [25]. In SIFU, the fundamental concept of unlearning is to identify the most recent global model using a bounded sensitivity metric calculated from historical contributions. Subsequently, the remaining clients retrain based on the identified model.

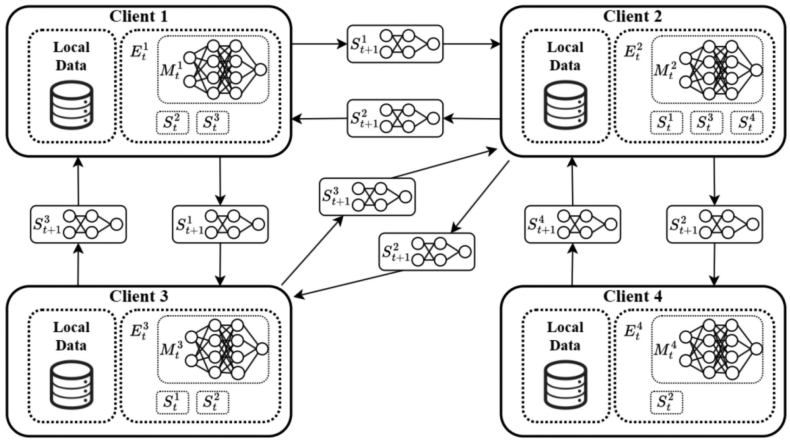

Fig. 8. An illustration of HDUS [141]. Operating without a central server, each client possesses their own neighboring distilled models, referred to as seed models. When a client requests to leave, the adjacent clients simply delete the seed model of the unlearned client. For predictions, an ensemble learning method is employed to combine the outputs of the primary model with those of the seed models.

Other approaches concentrate on unlearning without reliance on historical updates. In SFU [58], upon receiving an unlearning request, the server refines the global model using gradient information provided by the target client and representation matrix information provided by other clients. In KNOT [112], clients are grouped into clusters based on their training time and model sparsity. Clients within the same cluster collectively aggregate and update the global model asynchronously. When a client requests to leave, only clients within the same cluster require retraining. A similar structure is adopted in FedCIO [100] where clients are clustered according to their data distribution. Differing from conventional FL systems, HDUS [141] operates without a central server, as shown in Figure 8. Instead, each client possesses their own neighboring distilled models, referred to as seed models. When a client requests to leave, the adjacent clients simply delete the seed model of the unlearned client. For predictions, an ensemble learning method is employed to combine the outputs of the primary model with those of the seed models. Incentive mechanisms along with game theoretical analysis in FU systems is presented in References [19, 20, 68].

Client-aided unlearning inherently depends on the involvement of remaining clients and their updates, which can be a vulnerability in dynamic environments where client participation fluctuates. Additionally, this unlearning method can be slow, as it relies on all remaining clients, often resource-constrained devices, and is limited by the bandwidth of the FL system. This could lead to inefficiencies, particularly in cross-device scenarios with frequent client turnover or limited system resources.

4.2 Active Unlearning

"Active unlearning" denotes that the target client actively engages in the unlearning process and then has the option to either stay or leave, with or without verification (see Section 4.3 for more details on verification mechanisms). Given the direct access the target client possesses to the data to be unlearned, this approach exhibits greater potential as indicated by existing research. A comprehensive summary of active FU schemes is presented in Table 4.

4.2.1 Unlearn Partial Data. To unlearn the partial data of the target client, retraining is the most straightforward approach. To mitigate the computational cost of starting from scratch with

Table 3. A Summary of Passive FU Schemes

	Ref.	Who-Unlearn	Unlearn-What	Principle	Method	Verifie	Verify Method
Passive Unlearning	[54, 146]	Server	Target Client	Fine-tuning	Iteratively remove the contributions of the target client evaluated based on its historical gradient residuals.	NA	Accuracy-based metrics, unlearning time, MIA
	[8]	Server and Remaining clients	Target client	Model scrubbing	The server scrubs the model iteratively based on the estimation over historical gradients and global models, while the remaining clients periodically participate to eliminate accumulated estimation errors.	NA	Test error rate, backdoor attack, average computation/communication costs saving
	[75]	Server	Target client	Retraining	Remove nodes affected by the target client until reaching the leaf node. Then reconstruct the affected branches based on previously stored intermediate information.	NA	Accuracy-based metrics
	[129, 130]	Server	Target client	Multi-task unlearning	Unlearn by directly averaging the models of remaining clients, while avoiding forgetting by optimizing the knowledge distillation loss between unlearned model and previous global model.	NA	Accuracy-based metrics, backdoor attack
	[112]	Sever and Target client and Some remaining clients	Target client	Retraining	Divide clients into clusters for asynchronous aggregation. To unlearn some data, only clients in the same cluster are retrained.	Sever	Validation accuracy, deviation across recent validation accuracies
	[58]	Server and All clients	Target client	Gradient ascent	The server refines the global model with gradient ascent in a subspace based on the gradient provided by the target client and the representation matrix provided by the remaining clients.	NA	Backdoor attack
	[25]	Server and Remaining clients	Target client	Retraining	Find a historical global model based on a bounded sensitivity metric calculated based on clients' historical contributions, from where the remaining clients retrain.	NA	Number of retraining rounds, accuracy-based metrics
	[142]	Server and Remaining clients	Target client	Fine-tuning	Iteratively and selectively calibrate historical gradients to reconstruct the calibrated global model.	NA	Backdoor attack
	[141]	Remaining clients	Target Client	Model scrubbing	Each client retains neighboring distilled models, and predictions are obtained through an ensemble of the main model and seed model. To unlearn a target client, simply delete the seed model associated with that target client.	NA	Accuracy-based metrics
	[71]	Server and Remaining clients	Target client	Fine-tuning	Iteratively calibrate historical gradients to reconstruct the calibrated global model.	NA	Metrics, parameter deviation, MIA

(Continued)

Table 3. A Summary of Passive FU Schemes (Continued)

	Ref.	Who-Unlearn	Unlearn-What	Principle	Method	Verifie	Verify Method
Passive Unlearning	[97]	Server	Target client	Fine-tuning	Server aggregate the vectors from remaining clients representing their local clustering result.	Server	Global model convergence
	[28]	Server	Target client	Fine-tuning	The server then amplifies the gradients from the remaining clients and reduces the gradients of the target client.	Target client	Accuracy-based metrics
	[67]	Server and Remaining clients	Target client	Retraining	Retraining based on isolated shard and coded computing	NA	Accuracy, time, storage, MIA
	[120]	Remaining clients	Target client	Fine-tuning	Iteratively calibrate historical gradients to reconstruct the calibrated global model.	NA	Backdoor attack
	[26, 107]	Server	Target client	Fine-tuning	Calibrate historical gradients with penalty term based on projected gradients.	NA	Accuracy-based metrics, backdoor attack
	[41]	Server	Target client	Fine-tuning	Fine-tuning the model by subtracting target model updates.	NA	Accuracy-based metrics, time, CPU usage, memory
	[49]	Server	Target client	Fine-tuning	Calibrate historical gradients with guidance of estimated skew.	NA	Accuracy, backdoor attack
	[113]	Server	Target client and Partial data	Retraining	Strategic retraining based on the change of sampling probability.	NA	Accuracy, time, MIA

retraining, one solution is to roll back the global model to a state where it has not been significantly influenced by the target client. From this point, all FL clients can conduct the retraining process. For instance, in Exact-Fun [134], where FL models are quantized, when a client requests to leave, the client calculates a new model based on the remaining data. If the original model matches the new quantized model, signifying that the removal has no impact, the FL model remains unchanged. Otherwise, retraining is required to eliminate the influence of the unlearned data. In ViFLa [23], which is essentially a machine unlearning scheme, training samples are segmented into different groups, with each group representing an FL client. Hence, ViFLa can simulate an FU process in this context. The local model is trained using ring-based SQ-learning for LSTM, and weighted aggregation is determined by KL-attention scores. The historical model parameters represented by states over a ring are stored. To unlearn partial data, each client removes unlearned data and computes the new updates. Based on these new updates, the server identifies a previous state from which the remaining clients continue their training. A similar concept of identifying the optimal previous state for retraining is also present in SIFU [25], as discussed earlier in Section 4.1.2. In SCMA [97], a straightforward approach to unlearning partial data involves naive retraining. Each client maintains a vector representing its local clustering result, and these vectors are aggregated by the server to create a global clustering outcome. To unlearn partial data, SCMA entails each client calculating a new local vector, i.e., retraining, and then re-aggregating all vectors.

Fine-tuning and multi-task learning are popular approaches for FU as well. As an example, in FRAMU [105], the server aggregates fine-tuned local models and attention scores. Using these scores, it filters out irrelevant data points and updates the global model. The attention scores are acquired through local reinforcement learning applied to dynamic data. In FedLU [156], designed for FL over knowledge graphs where embeddings are aggregated instead of gradients as in standard FL, the unlearning of partial data is accomplished through iterative optimization of local

embeddings. This process follows the multi-task unlearning concept, involving unlearning over the local model and learning over the global model. In FedME^2 [133], clients engage in multi-task learning to optimize the loss of the local model, the loss from an MIA-like evaluation model, and a penalty term that accounts for the difference between the local model and the global model. In [115], unlearning is conducted by optimizing model performance on the remaining dataset while considering bias caused by the unlearned data.

Model scrubbing-based methods are commonly employed for unlearning partial data. In [76], the model scrubbing technique is applied to the target client to locally unlearn the partial data, involving Hessian matrix computations, with enhanced computational efficiency through an approximate diagonal empirical **Fisher Information Matrix** (**FIM**). In Forsaken [74], dummy gradients are computed to align the confidence vector of the unlearned model with that of a perfectly unlearned model. Forsaken+ [91] minimizes the distance between the posteriors of the data to be forgotten and those of non-member data for unlearning. FedAU [38] relies on the linear combination to approximate the unlearned model utilizing a pre-computed auxiliary model during the learning process. Reference [37] focus on feature unlearning by minimizing local feature sensitivity through model scrubbing. A similar approach of local unlearning followed by aggregation is described in CONFUSE [93] for multi-task unlearning at different levels. FFMU [9] treats data removals as perturbations on the dataset, employing **random smoothing** (**RS**) [15] to obtain a smoother model to simulate an unlearning process. In particular, FFMU aligns with the fundamental idea presented in PCMU [150], which involves randomized gradient smoothing combined with gradient quantization as follows.

$$S(\bar{G}) = \underset{c\in\{-1,0,1\}}{\operatorname{argmax}} \mathbb{P}_{\mathcal{D}}(Q(\bar{G}+\varepsilon)=c)$$

where $\mathcal{D} = \mathcal{N}\left(0,\sigma^2 I\right)$ is a Gaussian distribution, Q is a gradient quantization to map each dimension of the continuous gradient $G(x,y)\in\mathbb{R}^T$ over a discrete three-class space $\{-1,0,1\}$, for mimicking the classification in the randomized smoothing for certified robustness. S is a smooth version of Q, and returns whichever gradient classes Q^t is most likely to return when $\bar{G}$ is perturbed by noise ε. Extending FFMU from PCMU poses a challenge due to FL's privacy requirements, limiting server access to clients' local training data and affecting the certified data removal radius and budget in the global model. Therefore, by leveraging the theory of nonlinear functional analysis, the local MU models $g(x;q)$ in FFMU are reformulated as output functions of a Nemytskii operator $O(q)(x)$ where $q = Q(\bar{G})+\varepsilon$. In this way, the global unlearned model with bounded errors can maintain the certified radius and budget of data removals of the local unlearned models within a distance $\frac{(K-1)Cd}{\sqrt{2\pi}K\delta}$ (see Reference [9] for more details).

Some FU approaches are proposed with the use of synthetic data and gradient ascent principles. For instance, in UKRL[137], unlearning is conducted by training on perturbed unlearned data. In FedAF [63], shown in Figure 9, synthetic labels are generated for the data to be unlearned. A trusted third party creates random teacher models, and ensemble predictions from these models to provide synthetic labels for the unlearned data. Training is then conducted using this data with synthetic labels to achieve unlearning. Note that a multi-task approach is also employed in FedAF, where an additional task is introduced to retain the memory of the remaining data. In another work [1], which focuses on how adversaries can stealthily perform backdoor unlearning to evade server detection, the gradient ascent method is employed. The loss from the local benign dataset is utilized to constrain unbounded losses during the unlearning process based on gradient ascent.

Unlearning is also applicable in Bayesian federated learning systems. In Forget-SVGD [34], when a client requests to leave, the target client leverages the remaining data to compute the posterior

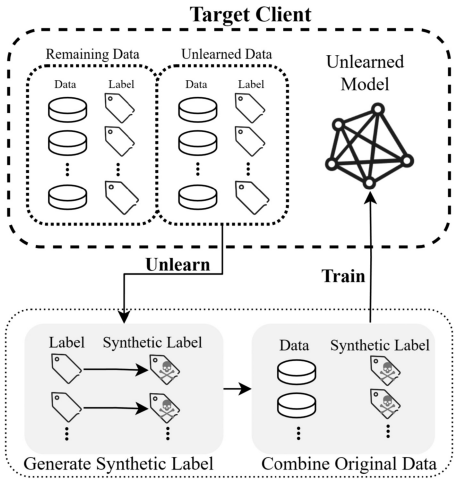

Fig. 9. An illustration of FedAF [63]. Synthetic labels are generated for the data to be unlearned. A trusted third party creates random teacher models, and ensemble predictions from these models to provide synthetic labels for the unlearned data. Training is then conducted using this data with synthetic labels to achieve unlearning.

probability, approximated through **variational inference** (**VI**), and performs an extra FL round for model updating. Similar unlearning methods are employed in [33, 36]. BVIFU [35] shares a core concept with Forget-SVGD, employing exponential family distributions in VI to approximate posterior probability. Unlike retraining-based unlearning in Bayesian FL, BFU [122] introduces a multi-task unlearning approach. It employs a parameter self-sharing method to balance between forgetting the unlearned data and remembering the knowledge learned by the original model, where probability distributions are approximated by a neural network.

In addition to the aforementioned approaches, the work in Reference [116] specializes in unlearning a specific type of partial data, particularly focusing on a category within the training dataset, i.e., a class. Typically initiated by the server, this unlearning process involves the application of quantization and pruning techniques. Specifically, the locally trained CNN model takes private images as input and produces feature map scores that assess the relationship between each channel and category. These scores are transmitted to the central server and aggregated into global feature map scores. The server utilizes TF-IDF to evaluate the relevance scores between channels and categories and creates a pruner to perform selective pruning on the most distinguishing channels of the target category. Subsequently, normal federated training proceeds with the exclusion of training data associated with the target category.

Limitations. Unlearning partial data through active unlearning is notably complex among all unlearning targets. It requires the removal of specific data while maintaining model performance on the remaining data. This complexity often results in more intricate algorithms, leading to increased computational and communication costs. Furthermore, this setting is susceptible to attacks based on over-unlearning, as identified in [44], where adversaries can exploit the unlearning process to enhance the effects of remaining poisoned data, thus facilitating poisoning attacks. This vulnerability underscores the need for careful consideration in implementing active partial data unlearning.

4.2.2 Unlearn Entire Client. As mentioned earlier, unlearning through gradient ascent is a versatile approach suitable for both partial data and target client unlearning. As detailed in

Reference [42], the target client employs gradient ascent to maximize the local loss, subject to constraints determined by the reference model provided by the remaining clients. Specifically, during FL training, a client's objective is to address the following optimization problem:

$$\min_{w} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} \mathcal{L}(w; (x_i, y_i))$$

where $\mathcal{L}(w; (x_i, y_i))$ represents the loss function, which calculates the prediction error for an individual example (x_i, y_i) from the dataset D, using the model parameters w. The unlearning method designed in Reference [42] is to reverse this learning process. That is, during unlearning, instead of learning model parameters that minimize the empirical loss, the client i strives to learn the model parameters to maximize the loss. Additionally, to prevent the process of gradient ascent from producing an arbitrary model similar to a random model, the average of the other clients' models, i.e., $w_{ref} = \frac{1}{N-1} \sum_{i \neq j} w_j$, is used as a reference model, and an ℓ_2-norm ball around the reference model is employed to limit the unbounded loss. Thus, during unlearning, the client solves the following optimization problem:

$$\max_{w: ||w - w_{ref}||_2 \leq \delta} \frac{1}{|D_i|} \sum_{(x_k, y_k) \in D_i} \mathcal{L}(w; (x_k, y_k))$$

Similarly to constraints, concurrent boost training on the remaining data [117] and reference models [3] can also be adopted as mitigation methods. To reduce the computation overhead, unlearning is conducted based on gradient ascent on distilled dataset [17]. Apart from the unlearning principle based on gradient ascent, 2F2L [55] and Appro-Fun [134] adopt model scrubbing methods similar to previous works such as [76] and [8]. To enhance computational efficiency, the complex task of Hessian matrix inversions is approximated using a pre-trained deep neural network and Taloy expansion.

Limitations. Unlearning an entire client through active learning poses limitations in scenarios where the unlearning request comes from the client to be unlearned. In such cases, the client must remain in the system for unlearning, conflicting with the common "request then leave" behavior in FL systems. Moreover, this method necessitates the unlearned client's continued participation alongside remaining clients, which is not cost-effective, especially in large-scale and resource-constrained FL systems.

4.3 Verification

In line with the description provided in VERIFI [28], the participant who requests unlearning, e.g., a target client or the server, is granted "the right to verify" (RTV). This means that the requester has the ability to actively verify the unlearning effect after the unlearning process is completed. This section will provide an overview of the verification mechanisms proposed in existing FU works.

4.3.1 Client-Side. To ensure the "right to verify," it is imperative that federated unlearning schemes provide clients with the ability to confirm the successful unlearning of their data. Regrettably, this aspect has received limited attention in existing FU literature. In EMA [47], the target client employs various metrics, including correctness, confidence, and negative entropy, to assess the performance of the audited dataset concerning the global model. These metrics are then ensembled to determine whether they meet a predefined threshold, serving as an indicator of whether the target client's data has been effectively unlearned.

Another noteworthy contribution in this domain is VERIFI [28]. VERIFI introduces two non-invasive verification methods that distinguish themselves from invasive techniques involving the injection of backdoors or watermarks, which manipulate the original data. In contrast,

Table 4. A Summary of Active FU Schemes

	Ref.	Who-Unlearn	Unlearn-What	Principle	Method	Verifie	Verify Method
Active Unlearning	[105]	Server and All clients	Partial data	Fine-tuning	The server aggregates local models and attention scores from all clients, based on which filters out unlearned data points and updates the global model.	Server	Global model convergence
	[156]	Server and All clients	Partial data	Fine-tuning and Multi-task unlearning	All clients iteratively optimize the local embedding based on mutual knowledge distillation following a multi-task style.	Server	Prediction results on knowledge
	[9]	Server	Partial data	Model scrubbing	Treat unlearned data as a perturbation on the whole dataset. Refine the global model by random smoothing.	Server	Certified budget of data removals
	[23]	Server and All clients	Partial data	Retraining	After each client removes unlearned data and calculates the new updates, the server identifies a prior state from which the remaining clients proceed with their training.	Server	Global model convergence
	[134]	All clients	Partial data	Retraining	The client calculates a new model based on the remaining data. If the original model matches the new quantized model, the FL model remains unchanged. Otherwise, retraining is required.	NA	Accuracy-based metrics, MIA, speed-up ratio
	[33, 34, 36]	Target client	Partial data	Retrain	Use variational inference to approximate Bayesian posterior probability. After the client requests to leave, the client uses the remaining data to reapproximation the posterior probability and execute an extra round of model upload	NA	Accuracy, Posterior Distribution
	[74]	Target clients	Partial data	Model scrubbing	Dummy gradients are computed to align confidence vectors of the unlearned model with that of a perfectly unlearned model.	NA	Forgetting rate
	[116]	Server and all clients	Partial data (a class)	Model scrubbing	The server assesses the relevance between channels and classes and establishes a pruner to selectively trim the most distinguishing channels of the target class.	NA	Accuracy-based, speed-up ratio, MIA
	[133]	All clients	Partial data	Multi-task unlearning	Engage in multi-task learning to optimize the loss of the local model, the loss from an MIA-like evaluation model, and a penalty from the difference between the local model and the global model.	NA	Convergence analysis, accuracy-based metrics, forgetting rate
	[91]	Target client	Partial data	Model Scrubbing	Target clients iteratively minimizes the distance between the posteriors of the data to be forgotten and those of non-member data for unlearning.	NA	Accuracy-based and efficiency-based metrics, MIA

(Continued)

Table 4. A Summary of Active FU Schemes (Continued)

	Ref.	Who-Unlearn	Unlearn-What	Principle	Method	Verifie	Verify Method
Active Unlearning	[63]	Target clients	Partial data	Multi-task unlearning	Synthetic labels are generated based on teacher ensembles for the data to be unlearned, and training is conducted using this data with synthetic labels to achieve unlearning.	NA	Accuracy, running time
	[122]	Target client	Partial data	Multi-task unlearning	Adopts a multi-task unlearning approach that utilizes a parameter self-sharing method to strike a balance between forgetting the unlearned data and retaining the remaining knowledge.	NA	Running time, model differences, accuracy-based metrics, backdoor attack
	[35]	Target client	Partial data	Retraining	Shares a core concept with [34], employing exponential family distributions in VI to approximate posterior probability.	NA	KL-divergence
	[1]	Target clients	Partial data	Gradient ascent	Follows the gradient ascent method, utilizing the loss from the local benign dataset to constrain unbounded losses.	NA	Accuracy-based metrics, backdoor attack
	[76]	Target client	Partial data	Model scrubbing	Scrubs the model based on the approximation of Hessian matrix using the remaining data.	NA	Running time, accuracy, model utility
	[97]	Target clients	Partial data	Retraining	Each client computes a new local vector, and these vectors are subsequently aggregated by the server.	Server	Global model convergence
	[55]	Target client	Target client	Model scrubbing	Scrubs the model based on the approximation of Hessian matrix using public server data.	NA	Accuracy-based metrics, MIA
	[42]	Target client	Target client	Gradient ascent	Target client computes the maximum empirical loss with the constraint of the reference model from remaining clients.	NA	Accuracy-based metrics, backdoor attack
	[38]	Server and all clients	Partial data	Model Scrubbing	Linear combination of the trained model and auxiliary model obtained during unlearning.	NA	Accuracy
	[117]	Server	Target client	Gradient ascent	Unlearning low-quality data with concurrent boost training with good-quality data	Server	Accuracy, loss, running time
	[145]	Target client	Partial data	Retraining	Retrain the model based on prune local models	NA	Accuracy, loss
	[17]	All clients	Target client or a class	Gradient ascent	Reverse training on the distilled dataset	NA	Accuracy, time, rounds, data size, MIA
	[115]	Server and all clients	Partial data	Multi-task unlearning	Optimize model performance on the remaining dataset while considering bias caused by the unlearned data.	NA	Backdoor attack, L2 distance, JS divergence, T-test
	[152]	Server and target clients	Partial data or a class	Fine-tuning	Fine-tuning based on randomly initialized degradation models	NA	Backdoor attack, accuracy
	[125]	All clients	Partial data (a feature)	Retraining	Rapid retraining using first-order method based on reinitialized model.	NA	Accuracy-based metrics

(Continued)

Table 4. A Summary of Active FU Schemes (Continued)

Ref.	Who-Unlearn	Unlearn-What	Principle	Method	Verifie	Verify Method
[134]	Server and target client	Target client	Model scrubbing	Achieving indistinguishability based on DP definition.	NA	Accuracy, loss, MIA, speed-up ratio
[137]	Server and target client	Target client	Synthetic data	Training on perturbed unlearned data.	NA	Accuracy
[37]	Server and target client	Target client	Model scrubbing	Local feature unlearning to minimize feature sensitivity.	NA	Sensitivity, bias, backdoor attack
[93]	Server and target client	Target client or partial data	Multi-task unlearning	Influence removal based on confusion Loss and performance recovery based on saliency map.	NA	Accuracy, MIA, backdoor attack

the verification methods proposed in VERIFI operate without modifying the data itself. These methods revolve around tracking a subset of unlearned data known as "markers," selected based on specific criteria. The criteria encompass two primary categories: (i) Forgettable Memory, where markers are identified as a representative subset incurring a high variance of local training loss, and (ii) Erroneous Memory, which designates markers as incorrectly predicted samples labeled as erroneous. By actively monitoring the unlearned model's performance on these markers, clients can effectively verify whether the unlearning process has successfully removed the unlearned data.

4.3.2 Server-Side. The outcome of the verification conducted by the server plays an important role in determining when to stop the unlearning process within the FU system. For instance, FRAMU [105] terminates unlearning when the difference between two consecutive global models becomes smaller than a predefined parameter. KNOT [112] concludes unlearning based on the required validation accuracy and the standard deviation across a recent history of such validation accuracies. ViFLa [23] and SCMA [97] end unlearning when the model converges. FedLU [156] relies on prediction results derived from knowledge, while FFMU [9] assesses whether data removals exceed a certified budget.

4.3.3 Verification Metrics. While the remaining FU works reviewed in this survey do not explicitly introduce verification mechanisms, the verification metrics employed in these works to assess unlearning performance could provide valuable insights for future research. We have summarized these metrics adopted in reviewed FU works in Table 3.

Accuracy-based metrics over unlearned data are the most commonly utilized metrics in the reviewed FU works. For example, they are used in works such as [1, 8, 25, 42, 55, 63, 71, 76, 122, 130, 133, 146, 156]. Metrics based on running time are employed to assess the efficiency of unlearning algorithms, as demonstrated in References [8, 25, 63, 71, 74, 76, 91, 97, 122, 133, 146]. Furthermore, some works rely on verification through backdoor attacks, as in [1, 8, 42, 58, 91, 130, 142], while others use membership inference attacks, as seen in References [55, 71, 91, 146]. The difference between the unlearned model and the retrained model is adopted to evaluate unlearning performance in References [36, 91, 122].

4.3.4 Limitations. By surveying the existing FU literature, we can make several observations regarding verification as follows:

— Only a few research studies on FU take "who-verify" into account, and the verification in almost all FU schemes remains at an experimental assessment level.

— Most FU methods rely on the assumption that verification is conducted by the server rather than the clients.
— There are no standard or widely adopted methods for proof of unlearning.

These observations indicate the need for research into "who-verify" and the development of efficient and robust verification methods conducted by different participants, especially client-verify methods. For instance, when considering MLaaS with unlearning services, clients must be allowed to verify if their data has been unlearned and its impact on the FL model has been removed. Only when the data removal adheres strictly to the specified unlearning request can the trustworthiness of the federated unlearning system be maintained.

4.4 Lessons Learned

In this section, we present the key lessons learned from our review of existing FU methods.

— "Who-unlearn" and "Who-verify": According to the proposed taxonomy, unlearning can be carried out by the participant who initiates the unlearning request (e.g., client or server) or by other participants, excluding the one who made the request. Similarly, verification can also be performed by clients or the server. However, we observe that the alignment between "Who-unlearn" and "Who-verify" is not optimal in FU literature. In other words, an FU system should allow the participants who raise the unlearning request to either conduct the unlearning or perform the verification themselves. This ensures that the unlearning results are credible to the participants who made the request.
— Selection of unlearning principles: It can be observed that different unlearning principles vary in their reliance on access to the training data. For instance, gradient ascent-based unlearning methods heavily rely on the unlearned data, while fine-tuning-based methods may rely only on the remaining data, and retraining-based methods are more flexible. Therefore, selecting appropriate unlearning principles that align with the FU scenarios concerning data access levels should be carefully considered.
— Structure of unlearning requests: It is challenging to determine the structure of unlearning requests from the existing FU literature. The underlying assumption that the unlearner has direct access to the unlearned data appears contradictory to the privacy foundation of FL systems. For example, if the unlearner is the server and the unlearned data is held by the clients, this creates a conflict. The lack of consideration for the structure of unlearning requests and their integration within the FL system may hinder the adoption of unlearning services in a federated setting.
— Proof of unlearning: As mentioned earlier, there are no standard or widely adopted metrics for proof of unlearning. To deploy unlearning as a service within MLaaS, it is crucial to establish a standard, either globally or within regional organizations, to guide the design of unlearning verification. Additionally, emphasis should be placed on verification by the entity that raises the unlearning requests.

5 FL-Tailored Optimizations, Limitations, and Applications

In addition to the primary challenges and solutions discussed in earlier sections, FU schemes face limitations due to the unique characteristics of the FL setting. These include (i) constrained resources, (ii) participant heterogeneity, and (iii) security and privacy threats. Furthermore, since FU systems involve additional unlearning processes compared with FL systems, new security and privacy threats arise. To address these issues, various optimization approaches and solutions have been proposed. In this section, we will delve into these limitations and concerns arising from the unique characteristics of the FL setting and explore the efforts made to address them. A summary of these FL-tailored optimization methods and solutions in FU is provided in Table 5.

Table 5. A Summary of FL-Tailored Optimization Methods in FU

Limitation		Optimization Method	Reference
Constrained Resources	Memory	Selective storage	[54, 71, 142]
		Compression	[67]
	Communication	Size reduction	[134]
		Rounds reduction	[23, 25, 134]
		Clustering	[83, 100, 112, 124]
	Computation	Approximation	[8, 55, 63, 76]
		Parallel computation	[9]
		Outsource computation	[8, 63]
		Pre-computation	[38]
		Dataset distillation	[17]
Participant Heterogeneity	Partitioned feature	Vertical FL	[16, 75, 125, 145]
	Variational representation	Knowledge distillation	[105, 115, 156]
	Non-IID distribution	Weighted aggregation	[23, 105, 112, 116]
		Knowledge distillation	[141, 156]
	Training capability	Client clustering	[112]
	Diverse payoff	Incentive mechanism	[19, 20, 68]
Security and Privacy Threats	Privacy-preservation	Indirect information	[19, 97, 141]
		PETs	[74, 75, 83, 87, 134, 145, 146]

5.1 Constrained Resources

In cross-device FL settings, FL clients are typically resource-constrained mobile devices that may drop out of the system at any time [56]. As shown in the FU workflow in Figure 4, federated unlearning is a subsequent process after federated learning. Therefore, the characteristic of resource-constrained participants exists in both FL and FU systems. Consequently, it is crucial to consider the resource requirements and consumption of FU schemes.

5.1.1 Memory. In many FU works, historical information is essential for facilitating the unlearning process. Nevertheless, this implies that the server needs to store a substantial volume of data, leading to significant memory consumption. This historical information can be gradients and global models [8, 25, 42, 71, 130, 142], gradient residuals [146], specific state [105], or some intermediate results [75].

Approaches to mitigation. Memory consumption reduction can be achieved by selectively storing historical information. For instance, in FedEraser [71], the server stores clients' gradients at specific intervals of FL rounds or based on the importance of gradients [54]. Similarly, in FRU [142], only important updates to clients' item embeddings are stored. In addition, by adopting coding-based techniques, the storage can be further compressed as demonstrated in Reference [67].

5.1.2 Communication. Unlearning in the FU system typically necessitates FL clients to transmit extra information, such as gradients, to the server to facilitate the process. For example, in FRAMU [105], clients additionally send attention scores to the server. In FedLU [156], loss information is transferred for mutual knowledge distillation between the server and clients. In SFU [58], the remaining clients need to send their representation matrix to the server. In Reference [42], the remaining clients' models are sent to the target client as references.

Approaches to mitigation. Reducing the communication cost can be achieved by minimizing the size of the model to be transferred, which may involve methods like quantization [134], and by

reducing the number of unlearning rounds. For retraining-based unlearning, the FU system may roll back the global model to a state where it has not been significantly influenced by the target client. From this point, all FL clients can conduct the retraining process [23, 25, 134], thus reducing the unlearning rounds and enhancing communication efficiency. Besides, clustering is used to divide FL users into groups, each with its own model. The final inference is determined by a majority vote from these sub-models. This method confines unlearning processes to individual clusters, eliminating the need for participation from all users, thus improving efficiency [83, 100, 112, 124].

5.1.3 Computation. FU often requires clients to engage in additional computational tasks compared with standard FL. These tasks can involve generating dummy gradients [74], conducting online reinforcement learning [105], seed model generation [141], or computing Hessian matrices [8, 55, 63, 76]. It's important to note that some of these computational tasks can be resource-intensive, which could pose challenges for deploying unlearning mechanisms in FL systems.

Approaches to mitigation. To enhance computational efficiency, approximation methods are commonly employed to accelerate certain components of FU algorithms. For example, the computation on the Hessian matrix can be approximated using techniques like a pre-trained deep neural network with Taylor expansion [55], the L-BFGS algorithm [8], or the FIM [63, 76]. Additionally, other optimizations are based on different FU structures. For instance, some works conduct training and unlearning simultaneously [9], similar to the approach used in Reference [150] in an MU setting. Orthogonally, computation tasks can be outsourced to a trusted third party, as demonstrated in Reference [63]. Another approach is transferring the majority of tasks to the server for estimation while executing only a small part of computation tasks for calibration [8], or reducing the computational tasks in the unlearning process by involving some pre-computation during the learning phase [38]. In Reference [17], dataset distillation is adopted to compress the size of the dataset while preserving the unlearning performance, hence reducing the computational overhead.

5.2 Participant Heterogeneity

In both FL and FU systems, clients exhibit heterogeneity in various aspects, encompassing differences in data structures, data distributions, such as vertical partitioned features [16, 75], variational data representations [105, 156], and the presence of Non-IID data [23, 105, 112, 116]. Furthermore, there are disparities in training capabilities on computational, communication, and memory, with some clients operating on resource-constrained mobile or IoT devices [112]. The existence of such diversity highlights the importance of developing heterogeneity-aware approaches for federated unlearning.

Approaches to mitigation. To address challenges associated with vertical partitioned features, certain vertical federated learning schemes are employed, as seen in works like [16, 75, 125, 145]. To handle Non-IID data distributions and variations in data representations, weighted aggregation techniques are commonly utilized, leveraging different metrics such as attention-based mechanisms [23, 105], TF-IDF [116], and model sparsity [112]. The introduction of knowledge distillation techniques helps mitigate issues arising from data heterogeneity, bias, and diverse model architectures [115, 141, 156]. Additionally, clustering based on local computational resources is considered to achieve asynchronous aggregation for FL, along with clustered retraining for FU [112]. Furthermore, incentive mechanisms can be adopted in FU systems to deal with diverse payoffs for different FU participants [19, 20, 68].

5.3 Security and Privacy Threats

Privacy and security issues in FU systems encompass those present in FL, such as the risk of information leakage and both targeted and untargeted attacks on ML models (see Section 3 for more

details). For instance, the leakage from gradients even allows the attacker to recover images with pixel-wise accuracy and texts with token-wise matching [155]. Methods to mitigate such a risk focus on privacy-preserving aggregation [80, 81, 153]. Additionally, recent research highlights that malicious clients can launch (i) untargeted poisoning attacks, which aim at slowing the learning process or reduce the global model's performance [77, 108], or (ii) targeted backdoor attacks, where a backdoor is embedded into the model, triggering malicious behavior under specific input conditions [48, 52, 60]. Such attacks can quickly degrade the global model's performance or implant backdoors within a few FL rounds, with effects lasting for many rounds, posing serious security risks [66, 90, 147].

Approaches to mitigation. To address these security and privacy concerns, mitigation strategies involve the use of indirect information, such as transmitting representative vectors instead of centroids in federated clustering [97], calculating clients' contributions using federated Shapley values [19], and generating predictions on the ensemble of seed models acquired through knowledge distillation [141]. Moreover, integrating **privacy-enhancing techniques** (**PETs**) into the FL-FU workflow can bolster security and privacy guarantees, such as employing secure random forest construction for secure random forest re-construction [75], secure aggregation [82] for privacy-preserving gradient sum-up [74, 83], secure two-party computation for privacy-preserving unlearning [87], homomorphic encryption for initialization [145], and differential privacy mechanisms [146] for rendering unlearned mode indistinguishable from the retrained one.

5.4 Applications for Enhanced Security

In addition to ensuring RTBF, federated unlearning has significant applications in enhancing the security and integrity of federated learning models. In the context of poisoning recovery, it enables the removal of maliciously inserted data from trained models, thus restoring their original accuracy and reliability [8]. For backdoor removal, federated unlearning is instrumental in eliminating hidden backdoors in FL models [1, 130, 130], which could otherwise be exploited for adversarial purposes. Additionally, it plays a crucial role in addressing data misuse in unauthorized training by enabling the removal of improperly used data or outdated data [23, 141], thereby ensuring compliance with standards and regulations. These applications underscore federated unlearning's importance in maintaining the trustworthiness and security of ML models.

5.5 Lessons Learned

In this section, we present the key lessons learned from our review of FL-tailored optimizations in existing FU methods.

— Tradeoffs of resource consumption: The interplay between memory, communication, and computation is complex. We observe independent efforts to optimize efficiency in each of these areas within existing FU approaches. However, there is a lack of combined consideration, which is crucial, especially for resource-constrained FL participants. The tradeoffs between memory, communication, and computation should be thoroughly investigated to achieve optimal results.

— Consideration of participant heterogeneity: We observe that a few studies consider the heterogeneity among FL participants, but this area still requires further exploration. For instance, when managing heterogeneity based on training ability, memory, communication, and computation should all be taken into account. Additionally, existing FU literature primarily addresses simple Non-IID settings with basic data representations. There is a need to investigate complex Non-IID data with other representations, such as graphs.

— Security and privacy: More studies on machine unlearning reveal that additional privacy leakage can occur in the unlearning setting compared with the learning process. In addition,

malicious unlearning, where attackers raise crafted unlearning requests to achieve goals such as degrading model performance or injecting backdoors, exists. However, there is a lack of extended investigation into these issues and the development of defense strategies in the federated setting.

6 Discussions and Promising Directions

In previous sections, we conducted a comprehensive survey of FU schemes. However, given the rapid evolution of FU schemes and their increasing deployment, numerous emerging challenges and open problems are awaiting further investigation. Many of these challenges necessitate additional properties and broader capabilities from FU schemes. In this section, we extend our discussion to encompass these challenges and present potential research directions, highlighting areas where FU schemes can further enhance their capabilities.

6.1 Privacy-Preserving FU

The majority of FU schemes reviewed in this survey heavily rely on gradient information from the target client or all clients. For instance, historical client models and updated client models are exposed to the server in various schemes [8, 71, 142, 146]. However, it has been highlighted that with only a client's model and the global model, an attacker, such as a malicious central server, can accurately reconstruct a client's data in a pixel-wise manner for images or token-wise matching for texts, as discussed in Reference [155]. To counteract this "deep leakage from gradients," **privacy-preserving techniques** (**PPT**), such as **Homomorphic Encryption** (**HE**) [29], **Multi-Party Computation** (**MPC**) [140], and **Differential Privacy** (**DP**) [21], can be integrated to aggregate clients' locally trained models in a privacy-preserving manner. However, it's important to note that this approach significantly impacts the performance of existing FU algorithms, as the server no longer has access to the gradient of the target client. Therefore, there is a critical need for privacy-preserving FU methods that enable unlearning while preserving clients' data privacy. Additionally, as previously mentioned, there are additional security and privacy risks introduced in MU systems due to information leakage from the differences between the original and unlearned models [12]. This potential information leakage must also be analyzed within a FU system, and corresponding defense mechanisms should be developed.

6.2 Verification and Proof of Unlearning

As elaborated in Section 4.3 and summarized in Table 3, it is unfortunate that this aspect regarding "who-verify" has received limited attention in existing FU literature. Given that in most real-life FL systems, unlearning requests are typically initiated by a specific target FL client, there should be a heightened emphasis on client-side verification that allows clients to verify if their data has been unlearned and its impact on the FL model has been removed. This approach not only enhances privacy guarantees [114] but also aligns with the marking-then-verification strategy outlined in [28], hence maintaining the trustworthiness of the federated unlearning system. Traditional invasive marking methods, including watermarking, fingerprinting, and backdoor attacks, manipulate the original data, potentially impacting the performance of the FL model. Consequently, the exploration of effective and non-invasive verification mechanisms is a critical area of research within the context of FU.

Apart from verification mechanisms, the development of "proof of unlearning" using applied cryptography such as **Zero-Knowledge Proofs** (**ZKPs**) or **Trusted Execution Environments** (**TEEs**) presents a compelling research area, particularly for environments where mentioned unlearning verification methods are impractical or trust is limited. This approach offers enhanced cryptographic security guarantees, ensuring more robust and verifiable federated unlearning in

sensitive or distrustful settings. Emphasizing cryptographic and hardware-based solutions marks a critical step forward in secure and trustworthy ML practices.

6.3 Emerging Threats in Unlearning

Due to the nature of unlearning, additional security and privacy risks are introduced in FU systems. For instance, the information can be leaked by the differences between the original and unlearned models [12, 27, 45, 88]. This could exacerbate client privacy issues if an attacker has access to the model before and after the unlearning. Furthermore, from a security perspective, various studies demonstrate that adversarial users can submit crafted unlearning requests with untargeted goals, such as degrading the utility of the unlearned model [44, 99, 151], or untargeted goals, such as injecting backdoors [18, 85, 99, 151]. These issues highlight potential vulnerabilities in FU schemes. Research needs to focus on developing robust defense mechanisms to mitigate these risks and ensure the integrity and security of FU systems.

6.4 Awareness of Client-Dynamics

The process of federated unlearning introduces significantly more non-determinism compared to centralized machine unlearning. This increased complexity arises from the random selection of clients and data for global aggregation and local training in each round, as well as the presence of potentially numerous dropped and newly joined clients. The unlearning process becomes even more challenging when considering the need to recall past clients for unlearning and retraining. This is particularly difficult for more complex FU schemes, such as privacy-preserving FU, where the integration of PETs must also provide resilience to dynamic client participation. Addressing these dynamic challenges requires the development of client-dynamics-aware FU algorithms. Such algorithms must be capable of adapting to the fluid nature of client involvement in FU, ensuring the integrity and effectiveness of the unlearning process even as clients frequently join or leave the network. This area of research is crucial for the advancement of FU, aiming to create robust solutions that maintain high standards of privacy and efficiency despite the inherent non-determinism of federated environments.

6.5 Domain-Specific Applications

Machine unlearning techniques are employed in diverse scenarios, such as LLMs [57, 78, 98, 111], recommendation systems [10, 62, 64], and specific application scenarios such as health [22, 24, 154], IoT and blockchain [69, 72, 119, 143, 157], HAR [11], and metaverse and digital twin [50, 118], to adhere to data privacy and compliance objectives for RTBF. In LLMs, unlearning helps ensure that sensitive or outdated information can be effectively removed, maintaining user privacy and data accuracy. For recommendation systems, unlearning allows for the deletion of user-specific data upon request, thereby enhancing user trust and compliance with privacy regulations. These techniques are also utilized in **graph neural networks (GNNs)** [13, 14, 128], addressing privacy concerns in GNNs [126, 127, 144] to ensure data accuracy and relevance, and knowledge graphs to erase specific knowledge [70]. These applications demonstrate the versatility and critical role of machine unlearning in various technological domains. However, these domain-specific MU applications have yet to be extensively adapted to FU in federated settings. This gap underscores the potential for expanding the scope and applicability of FU strategies to these areas, encouraging further research into adapting these unlearning techniques for federated learning environments.

6.6 Fairness and Explainability

Researching fairness and explainability in FU algorithms is essential, given the intricacies of ML and the distributed nature of FL. For instance, overlapping data among different FL clients is a

common scenario. Unlearning data in such overlaps might fulfill the unlearning request from one client but could adversely affect the performance of other clients sharing that data. Furthermore, unlearning in FL introduces an extra layer of complexity, complicating the understanding of the model's alterations and their impact on the system as a whole. Addressing these issues is crucial to ensure transparency, trustworthiness, and adherence to regulations. Bridging the knowledge gap in this research field is imperative for improved decision-making interpretation in distributed AI systems. This is especially vital in industries where the processes of learning and unlearning carry profound ethical and legal connotations. Additionally, unlearning can be utilized as a method to enhance fairness in FL, which is often a challenge due to the Non-IID nature of the data across different clients. By selectively unlearning biased or unfair data contributions, the overall model can be adjusted to provide more equitable outcomes, addressing the inherent discrepancies that arise from Non-IID data distributions. Research in this field is promising as unlearning not only improves the fairness of the FL model but also ensures that the model's performance is more consistent and reliable across diverse data sources.

6.7 Integration with MLaaS

A key future direction for FU lies in its integration with MLaaS. This involves addressing several critical challenges. First, the structure of unlearning requests needs to be carefully designed to meet the stringent privacy requirements of FL. Effective protocols must ensure that unlearning processes do not compromise client privacy. Second, the system must efficiently handle multiple unlearning requests. This includes developing strategies to manage these requests in a scalable and responsive manner. Third, maintaining the **quality of service** (**QoS**) in MLaaS is essential, particularly regarding throughput and privacy guarantees. One challenge is that executing unlearning requests might necessitate halting inference services, impacting QoS. Conversely, not performing unlearning contravenes the RTBF regulation. Therefore, future research should focus on creating mechanisms that respect RTBF while minimizing service disruptions. This would enable MLaaS to deliver reliable, privacy-conscious, and high-performance ML services.

7 Conclusions

In conclusion, this survey has made remarkable contributions to the field of federated unlearning. We began by meticulously formalizing the targets and challenges of federated unlearning and introducing an innovative unified federated unlearning workflow. We then derived a novel taxonomy for existing federated unlearning methods, based on crucial factors such as who initiates the unlearning, what precisely needs to be unlearned, and how to effectively verify the unlearning results in federated settings. Furthermore, we thoroughly explored various optimizations tailored to federated learning and provided a critical examination of their limitations. Through these comprehensive efforts, we have gained profound insights into the current challenges in federated unlearning and have outlined promising research directions for the future. This survey stands as an invaluable and insightful resource for researchers and practitioners, significantly advancing the rapidly evolving field of federated unlearning.

References

[1] Manaar Alam, Hithem Lamri, and Michail Maniatakos. 2023. Get Rid Of Your Trail: Remotely erasing backdoors in federated learning. CoRR abs/2304.10638, (2023). DOI: https://doi.org/10.48550/ARXIV.2304.10638

[2] Eugene Bagdasaryan and Vitaly Shmatikov. 2021. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security'21)*. 1505–1521.

[3] Hasin Bano, Muhammad Ameen, Muntazir Mehdi, Amaad Hussain, and Pengfei Wang. 2023. Federated unlearning and server right to forget: Handling unreliable client contributions. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*. Springer, 393–410.

[4] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. 2020. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1253–1269.

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[6] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP'21)*. IEEE, 141–159.

[7] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21–25, 2021*.

[8] Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. 2023. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP'23)*. IEEE, 1366–1383.

[9] Tianshi Che, Yang Zhou, Zijie Zhang, Lingjuan Lyu, Ji Liu, Da Yan, Dejing Dou, and Jun Huan. 2023. Fast federated machine unlearning with nonlinear functional theory. In *International Conference on Machine Learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 4241–4268.

[10] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*. 2768–2777.

[11] Kongyang Chen, Dongping Zhang, Yaping Chai, Weibin Zhang, Shaowei Wang, and Jiaxing Shen. 2024. Federated unlearning for human activity recognition. CoRR abs/2404.03659, (2024). DOI: https://doi.org/10.48550/ARXIV.2404.03659

[12] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 896–911.

[13] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 499–513.

[14] Eli Chien, Chao Pan, and Olgica Milenkovic. 2022. Certified graph unlearning. CoRR abs/2206.09140, (2022). DOI: https://doi.org/10.48550/ARXIV.2206.09140

[15] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*. PMLR, 1310–1320.

[16] Zihao Deng, Zhaoyang Han, Chuan Ma, Ming Ding, Long Yuan, Chunpeng Ge, and Zhe Liu. 2023. Vertical federated unlearning on the logistic regression model. *Electronics* 12, 14 (2023), 3182.

[17] Akash Dhasade, Yaohong Ding, Song Guo, Anne-Marie Kermarrec, Martijn de Vos, and Leijie Wu. 2023. QuickDrop: Efficient federated unlearning by integrated dataset distillation. CoRR abs/2311.15603, (2023). DOI: https://doi.org/10.48550/ARXIV.2311.15603

[18] Jimmy Z. Di, Jack Douglas, Jayadev Acharya, Gautam Kamath, and Ayush Sekhari. 2022. Hidden poison: Machine unlearning enables camouflaged poisoning attacks. In *NeurIPS ML Safety Workshop*.

[19] Ningning Ding, Zhenyu Sun, Ermin Wei, and Randall Berry. 2023. Incentive mechanism design for federated learning and unlearning. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 11–20.

[20] Ningning Ding, Ermin Wei, and Randall Berry. 2024. Strategic data revocation in federated unlearning. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE.

[21] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3–4 (2014), 211–407. DOI: https://doi.org/10.1561/0400000042

[22] Khaoula ElBedoui. 2023. ECG classifiction based on federated unlearning. In *2023 International Symposium on Networks, Computers and Communications (ISNCC'23)*. IEEE, 1–5.

[23] Jiamin Fan, Kui Wu, Yang Zhou, Zhengan Zhao, and Shengqiang Huang. 2023. Fast model update for iot traffic anomaly detection with machine unlearning. *IEEE Internet Things J.* 10, 10 (2023), 8590–8602. DOI: https://doi.org/10.1109/JIOT.2022.3214840

[24] Yann Fraboni, Lucia Innocenti, Michela Antonelli, Richard Vidal, Laetitia Kameni, Sebastien Ourselin, and Marco Lorenzi. 2023. Validation of federated unlearning on collaborative prostate segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 322–333.

[25] Yann Fraboni, Martin Van Waerebeke, Kevin Scaman, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. 2024. SIFU: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In

International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain (Proceedings of Machine Learning Research), PMLR, 3457–3465. Retrieved from https://proceedings.mlr.press/v238/fraboni24a.html

[26] Chaohao Fu, Weijia Jia, and Na Ruan. 2024. Client-free federated unlearning via training reconstruction with anchor subspace calibration. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'24)*. IEEE, 9281–9285.

[27] Ji Gao, Sanjam Garg, Mohammad Mahmoody, and Prashant Nalini Vasudevan. 2022. Deletion inference, reconstruction, and compliance in machine (un)learning. *Proc. Priv. Enhancing Technol.* 2022, 3 (2022), 415–436. DOI: https://doi.org/10.56553/POPETS-2022-0079

[28] Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. 2022. VeriFi: Towards verifiable federated unlearning. CoRR abs/2205.12709, (2022). DOI: https://doi.org/10.48550/ARXIV.2205.12709

[29] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*. 169–178.

[30] Antonio Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. 2019. Making AI forget You: Data deletion in machine learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 3513–3526. Retrieved from https://proceedings.neurips.cc/paper/2019/hash/cb79f8fa58b91d3af6c9c991f63962d3-Abstract.html

[31] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 383–398.

[32] Eric Goldman. 2020. An introduction to the california consumer privacy act (ccpa). *Santa Clara Univ. Legal Studies Research Paper* (2020).

[33] Jine Gong. 2023. Bayesian learning and unlearning in distributed wireless network. phdthesis. Korea Advanced Institute of Science. Retrieved from https://koasas.kaist.ac.kr/handle/10203/309063

[34] Jinu Gong, Joonhyuk Kang, Osvaldo Simeone, and Rahif Kassab. 2022. Forget-svgd: Particle-based Bayesian federated unlearning. In *2022 IEEE Data Science and Learning Workshop (DSLW'22)*. IEEE, 1–6.

[35] Jinu Gong, Osvaldo Simeone, and Joonhyuk Kang. 2021. Bayesian variational federated learning and unlearning in decentralized networks. In *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC'21)*. IEEE, 216–220.

[36] Jinu Gong, Osvaldo Simeone, and Joonhyuk Kang. 2022. Compressed particle-based federated bayesian learning and unlearning. *IEEE Communications Letters* 27, 2 (2022), 556–560.

[37] Hanlin Gu, WinKent Ong, Chee Seng Chan, and Lixin Fan. 2024. Ferrari: federated feature unlearning via optimizing feature sensitivity. CoRR abs/2405.17462, (2024). DOI: https://doi.org/10.48550/ARXIV.2405.17462

[38] Hanlin Gu, Gongxi Zhu, Jie Zhang, Xinyuan Zhao, Yuxing Han, Lixin Fan, and Qiang Yang. 2024. Unlearning during Learning: An efficient federated machine unlearning method. In *33nd International Joint Conference on Artificial Intelligence (IJCAI'24)*.

[39] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying vulnerabilities in the machine learning model supply chain. CoRR abs/1708.06733, (2017). Retrieved from http://arxiv.org/abs/1708.06733

[40] Jiale Guo, Ziyao Liu, Kwok-Yan Lam, Jun Zhao, and Yiqiang Chen. 2021. Privacy-enhanced federated learning with weighted aggregation. In *Security and Privacy in Social Networks and Big Data: 7th International Symposium, SocialSec 2021, Fuzhou, China, November 19–21, 2021, Proceedings 7*. Springer, 93–109.

[41] Xintong Guo, Pengfei Wang, Sen Qiu, Wei Song, Qiang Zhang, Xiaopeng Wei, and Dongsheng Zhou. 2024. FAST: Adopting federated unlearning to eliminating malicious terminals at server side. *IEEE Trans. Netw. Sci. Eng.* 11, 2 (2024), 2289–2302. DOI: https://doi.org/10.1109/TNSE.2023.3343117

[42] Anisa Halimi, Swanand Ravindra Kadhe, Ambrish Rawat, and Nathalie Baracaldo Angel. 2022. Federated unlearning: How to efficiently erase a client in FL?. In *International Conference on Machine Learning*.

[43] Ling Han, Nanqing Luo, Hao Huang, Jing Chen, and Mary-Anne Hartley. 2024. Towards independence criterion in machine unlearning of features and labels. CoRR abs/2403.08124, (2024). DOI: https://doi.org/10.48550/ARXIV.2403.08124

[44] Hongsheng Hu, Shuo Wang, Jiamin Chang, Haonan Zhong, Ruoxi Sun, Shuang Hao, Haojin Zhu, and Minhui Xue. 2023. A duty to forget, a right to be assured? Exposing vulnerabilities in machine unlearning services. In *NDSS*. arXiv:2309.08230. Retrieved from https://arxiv.org/abs/2309.08230

[45] Hongsheng Hu, Shuo Wang, Tian Dong, and Minhui Xue. 2024. Learn what you want to unlearn: Unlearning inversion attacks against machine unlearning. In *2024 IEEE Symposium on Security and Privacy (SP'24)*.

[46] Yuke Hu, Jian Lou, Jiaqi Liu, Feng Lin, Zhan Qin, and Kui Ren. 2024. ERASER: Machine unlearning in MLaaS via an inference serving-aware approach. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*.

[47] Yangsibo Huang, Chun-Yin Huang, Xiaoxiao Li, and Kai Li. 2023. A dataset auditing method for collaboratively trained machine learning models. *IEEE Trans. Medical Imaging* 42, 7 (2023), 2081–2090. DOI: https://doi.org/10.1109/TMI.2022.3220706

[48] Yujin Huang, Terry Yue Zhuo, Qiongkai Xu, Han Hu, Xingliang Yuan, and Chunyang Chen. 2023. Training-free lexical backdoor attacks on language models. In *Proceedings of the ACM Web Conference 2023*. 2198–2208.

[49] Thanh Trung Huynh, Trong Bang Nguyen, Phi Le Nguyen, Thanh Tam Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. 2024. Fast-FedUL: A training-free federated unlearning with provable skew resilience. CoRR abs/2405.18040, (2024). DOI: https://doi.org/10.48550/ARXIV.2405.18040

[50] Anik Islam, Hadis Karimipour, Thippa Reddy Gadekallu, and Yaodong Zhu. 2024. A federated unlearning-based secure management scheme to enable automation in smart consumer electronics facilitated by digital twin. *IEEE Transactions on Consumer Electronics* (2024), 1–1. DOI: https://doi.org/10.1109/TCE.2024.3396723

[51] Hitomi Iwase. 2019. Overview of the act on the protection of personal information. *Eur. Data Prot. L. Rev.* 5 (2019), 92.

[52] Najeeb Moharram Jebreel and Josep Domingo-Ferrer. 2023. FL-Defender: Combating targeted attacks in federated learning. *Knowl. Based Syst.* 260 (2023), 110178. DOI: https://doi.org/10.1016/J.KNOSYS.2022.110178

[53] Hyejun Jeong, Shiqing Ma, and Amir Houmansadr. 2024. SoK: Challenges and opportunities in federated unlearning. CoRR abs/2403.02437, (2024). DOI: https://doi.org/10.48550/ARXIV.2403.02437

[54] Yu Jiang, Jiyuan Shen, Ziyao Liu, Chee Wei Tan, and Kwok-Yan Lam. 2024. Towards efficient and certified recovery from poisoning attacks in federated learning. CoRR abs/2401.08216, (2024). DOI: https://doi.org/10.48550/ARXIV.2401.08216

[55] Ruinan Jin, Minghui Chen, Qiong Zhang, and Xiaoxiao Li. 2023. Forgettable federated linear learning with certified data removal. CoRR abs/2306.02216, (2023). DOI: https://doi.org/10.48550/ARXIV.2306.02216

[56] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* 14, 1–2 (2021), 1–210. DOI: https://doi.org/10.1561/2200000083

[57] Vinayshekhar Bannihatti Kumar, Rashmi Gangadharaiah, and Dan Roth. 2023. Privacy adhering machine un-learning in NLP. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 - Findings, Nusa Dua, Bali, November 1-4, 2023*, Association for Computational Linguistics, 268–277. DOI: https://doi.org/10.18653/V1/2023.FINDINGS-IJCNLP.25

[58] Guanghao Li, Li Shen, Yan Sun, Yue Hu, Han Hu, and Dacheng Tao. 2023. Subspace based Federated Unlearning. CoRR abs/2302.12448, (2023). DOI: https://doi.org/10.48550/ARXIV.2302.12448

[59] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. 2020. Learning to detect malicious clients for robust federated learning. CoRR abs/2002.00211, (2020). Retrieved from https://arxiv.org/abs/2002.00211

[60] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 3123–3140.

[61] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. 2020. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2020), 2088–2105.

[62] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Junlin Liu, and Jun Wang. 2024. Making recommender systems forget: Learning and unlearning for erasable recommendation. *Knowl. Based Syst.* 283 (2024), 111124. DOI: https://doi.org/10.1016/J.KNOSYS.2023.111124

[63] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, and Jiaming Zhang. 2023. Federated Unlearning via Active Forgetting. CoRR abs/2307.03363, (2023). DOI: https://doi.org/10.48550/ARXIV.2307.03363

[64] Yuyuan Li, Chaochao Chen, Xiaolin Zheng, Yizhao Zhang, Biao Gong, Jun Wang, and Linxun Chen. 2023. Selective and collaborative influence function for efficient recommendation unlearning. *Expert Syst. Appl.* 234 (2023), 121025. DOI: https://doi.org/10.1016/J.ESWA.2023.121025

[65] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2024. Backdoor learning: A survey. *IEEE Trans. Neural Networks Learn. Syst.* 35, 1 (2024), 5–22. DOI: https://doi.org/10.1109/TNNLS.2022.3182979

[66] Yinshan Li, Hua Ma, Zhi Zhang, Yansong Gao, Alsharif Abuadbba, Minhui Xue, Anmin Fu, Yifeng Zheng, Said F. Al-Sarawi, and Derek Abbott. 2024. NTD: Non-transferability enabled deep learning backdoor detection. *IEEE Trans. Inf. Forensics Secur.* 19 (2024), 104–119. DOI: https://doi.org/10.1109/TIFS.2023.3312973

[67] Yijing Lin, Zhipeng Gao, Hongyang Du, Dusit Niyato, Gui Gui, Shuguang Cui, and Jinke Ren. 2024. Scalable federated unlearning via isolated and coded sharding. arXiv:2401.15957. Retrieved from https://arxiv.org/abs/2401.15957

[68] Yijing Lin, Zhipeng Gao, Hongyang Du, Dusit Niyato, Jiawen Kang, and Xiaoyuan Liu. 2024. Incentive and dynamic client selection for federated unlearning. In *Proceedings of the ACM on Web Conference 2024*. 2936–2944.

[69] Yijing Lin, Zhipeng Gao, Hongyang Du, Jinke Ren, Zhiqiang Xie, and Dusit Niyato. 2024. Blockchain-enabled trustworthy federated unlearning. CoRR abs/2401.15917, (2024). DOI: https://doi.org/10.48550/ARXIV.2401.15917

[70] Bingchen Liu and Yuanyuan Fang. 2024. Federated knowledge graph unlearning via diffusion model. CoRR abs/2403.08554, (2024). DOI: https://doi.org/10.48550/ARXIV.2403.08554

[71] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. 2021. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS'21)*. IEEE, 1–10.

[72] Xiao Liu, Mingyuan Li, Xu Wang, Guangsheng Yu, Wei Ni, Lixiang Li, Haipeng Peng, and Ren Ping Liu. 2024. Decentralized federated unlearning on blockchain. CoRR abs/2402.16294, (2024). DOI: https://doi.org/10.48550/ARXIV.2402.16294

[73] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS'18)*. Internet Soc.

[74] Yang Liu, Zhuo Ma, Ximeng Liu, and Jianfeng Ma. 2020. Learn to Forget: User-level memorization elimination in federated learning. CoRR abs/2003.10933, (2020). Retrieved from https://arxiv.org/abs/2003.10933

[75] Yang Liu, Zhuo Ma, Yilong Yang, Ximeng Liu, Jianfeng Ma, and Kui Ren. 2021. Revfrf: Enabling cross-domain random forest training with revocable federated learning. *IEEE Transactions on Dependable and Secure Computing* 19, 6 (2021), 3671–3685.

[76] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. 2022. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1749–1758.

[77] Yi Liu, Xingliang Yuan, Ruihui Zhao, Cong Wang, Dusit Niyato, and Yefeng Zheng. 2020. Poisoning semi-supervised federated learning via unlabeled data: Attacks and defenses. arXiv:2012.04432. Retrieved from https://arxiv.org/abs/2012.04432

[78] Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. Towards safer large language models through machine unlearning. In *Findings of the Association for Computational Linguistics: ACL 2024*.

[79] Zheyuan Liu, Guangyao Dou, Eli Chien, Chunhui Zhang, Yijun Tian, and Ziwei Zhu. 2024. Breaking the trilemma of privacy, utility, and efficiency via controllable machine unlearning. In *Proceedings of the ACM on Web Conference 2024 (WWW'24)*, Singapore, ACM, 1260–1271. DOI: https://doi.org/10.1145/3589334.3645669

[80] Ziyao Liu, Jiale Guo, Kwok-Yan Lam, and Jun Zhao. 2023. Efficient dropout-resilient aggregation for privacy-preserving machine learning. *IEEE Trans. Inf. Forensics Secur.* 18 (2023), 1839–1854. DOI: https://doi.org/10.1109/TIFS.2022.3163592

[81] Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao. 2022. Privacy-preserving aggregation in federated learning: A survey. CoRR abs/2203.17005, (2022). DOI: https://doi.org/10.48550/ARXIV.2203.17005

[82] Ziyao Liu, Jiale Guo, Wenzhuo Yang, Jiani Fan, Kwok-Yan Lam, and Jun Zhao. 2024. Dynamic user clustering for efficient and privacy-preserving federated learning. *IEEE Transactions on Dependable and Secure Computing* (2024), 1–12. DOI: https://doi.org/10.1109/TDSC.2024.3355458

[83] Ziyao Liu, Yu Jiang, Weifeng Jiang, Jiale Guo, Jun Zhao, and Kwok-Yan Lam. 2024. Guaranteeing data privacy in federated unlearning with dynamic user participation. CoRR abs/2406.00966, (2024). DOI: https://doi.org/10.48550/ARXIV.2406.00966

[84] Ziyao Liu, Hsiao-Ying Lin, and Yamin Liu. 2023. Long-term privacy-preserving aggregation with user-dynamics for federated learning. *IEEE Trans. Inf. Forensics Secur.* 18 (2023), 2398–2412. DOI: https://doi.org/10.1109/TIFS.2023.3266919

[85] Zihao Liu, Tianhao Wang, Mengdi Huai, and Chenglin Miao. 2024. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 14115–14123.

[86] Ziyao Liu, Huanyi Ye, Chen Chen, and Kwok-Yan Lam. 2024. Threats, attacks, and defenses in machine unlearning: A survey. arXiv:2403.13682. Retrieved from https://arxiv.org/abs/2403.13682

[87] Ziyao Liu, Huanyi Ye, Yu Jiang, Jiyuan Shen, Jiale Guo, Ivan Tjuawinata, and Kwok-Yan Lam. 2024. Privacy-preserving federated unlearning with certified client removal. CoRR abs/2404.09724, (2024). DOI: https://doi.org/10.48550/ARXIV.2404.09724

[88] Zhaobo Lu, Hai Liang, Minghao Zhao, Qingzhe Lv, Tiancai Liang, and Yilei Wang. 2022. Label-only membership inference attacks on machine unlearning without dependence of posteriors. *International Journal of Intelligent Systems* 37, 11 (2022), 9424–9441.

[89] Lingjuan Lyu, Han Yu, and Qiang Yang. 2020. Threats to Federated Learning: A Survey. CoRR abs/2003.02133, (2020). Retrieved from https://arxiv.org/abs/2003.02133

[90] Hua Ma, Huming Qiu, Yansong Gao, Zhi Zhang, Alsharif Abuadbba, Minhui Xue, Anmin Fu, Jiliang Zhang, Said F. Al-Sarawi, and Derek Abbott. 2024. Quantization backdoors to deep learning commercial frameworks. *IEEE Trans. Dependable Secur. Comput.* 21, 3 (2024), 1155–1172. DOI: https://doi.org/10.1109/TDSC.2023.3271956

[91] Zhuo Ma, Yang Liu, Ximeng Liu, Jian Liu, Jianfeng Ma, and Kui Ren. 2023. Learn to forget: Machine unlearning via neuron masking. *IEEE Trans. Dependable Secur. Comput.* 20, 4 (2023), 3194–3207. DOI: https://doi.org/10.1109/TDSC.2022.3194884

[92] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[93] Syed Irfan Ali Meerza, Amir Sadovnik, and Jian Liu. 2024. ConFUSE: Confusion-based federated unlearning with salience exploration. In *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, IEEE, 1–6.

[94] Thai-Hung Nguyen, Hong-Phuc Vu, Dung Thuy Nguyen, Tuan Minh Nguyen, Khoa D. Doan, and Kok-Seng Wong. 2024. Empirical study of federated unlearning: Efficiency and effectiveness. In *Asian Conference on Machine Learning*. PMLR, 959–974.

[95] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. CoRR abs/2209.02299, (2022). DOI: https://doi.org/10.48550/ARXIV.2209.02299

[96] Jorge Nocedal. 1980. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation* 35, 151 (1980), 773–782.

[97] Chao Pan, Jin Sima, Saurav Prakash, Vishal Rana, and Olgica Milenkovic. 2022. Machine unlearning of federated clusters. In *The Eleventh International Conference on Learning Representations*.

[98] Martin Pawelczyk, Seth Neel, and Himabindu Lakkaraju. 2023. In-context unlearning: Language models as few shot unlearners. CoRR abs/2310.07579, (2023). DOI: https://doi.org/10.48550/ARXIV.2310.07579

[99] Wei Qian, Chenxu Zhao, Wei Le, Meiyi Ma, and Mengdi Huai. 2023. Towards understanding and enhancing robustness of deep learning models against malicious unlearning attacks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1932–1942.

[100] Hongyu Qiu, Yongwei Wang, Yonghui Xu, Lizhen Cui, and Zhiqi Shen. 2023. FedCIO: Efficient exact federated unlearning with clustering, isolation, and one-shot aggregation. In *2023 IEEE International Conference on Big Data (BigData'23)*. IEEE, 5559–5568.

[101] Youyang Qu, Xin Yuan, Ming Ding, Wei Ni, Thierry Rakotoarivelo, and David B. Smith. 2023. Learn to Unlearn: A survey on machine unlearning. CoRR abs/2305.07512, (2023). DOI: https://doi.org/10.48550/ARXIV.2305.07512

[102] General Data Protection Regulation. 2018. General data protection regulation (GDPR). *Intersoft Consulting, Accessed in October* 24, 1 (2018).

[103] Nicolò Romandini, Alessio Mora, Carlo Mazzocca, Rebecca Montanari, and Paolo Bellavista. 2024. Federated unlearning: A survey on methods, design guidelines, and evaluation metrics. CoRR abs/2401.05146, (2024). DOI: https://doi.org/10.48550/ARXIV.2401.05146

[104] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11957–11965.

[105] Thanveer Shaik, Xiaohui Tao, Lin Li, Haoran Xie, Taotao Cai, Xiaofeng Zhu, and Qing Li. 2023. FRAMU: Attention-based machine unlearning using federated reinforcement learning. CoRR abs/2309.10283, (2023). DOI: https://doi.org/10.48550/ARXIV.2309.10283

[106] Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. 2023. Exploring the landscape of machine Unlearning: A comprehensive survey and taxonomy. CoRR abs/2305.06360, (2023). DOI: https://doi.org/10.48550/ARXIV.2305.06360

[107] Jiaqi Shao, Tao Lin, Xuanyu Cao, and Bing Luo. 2024. Federated Unlearning: a Perspective of Stability and Fairness. CoRR abs/2402.01276, (2024). DOI: https://doi.org/10.48550/ARXIV.2402.01276

[108] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. 2022. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP'22)*. IEEE, 1354–1371.

[109] Shiqi Shen, Shruti Tople, and Prateek Saxena. 2016. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 508–519.

[110] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP'17)*. IEEE, 3–18.

[111] Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. 2023. Knowledge unlearning for LLMs: tasks, methods, and challenges. CoRR abs/2311.15766, (2023). DOI: https://doi.org/10.48550/ARXIV.2311.15766

[112] Ningxin Su and Baochun Li. 2023. Asynchronous federated unlearning. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 1–10.

[113] Youming Tao, Cheng-Long Wang, Miao Pan, Dongxiao Yu, Xiuzhen Cheng, and Di Wang. 2024. Communication efficient and provable federated unlearning. *Proc. VLDB Endow.* 17, 5 (2024), 1119–1131.

[114] Fei Wang, Baochun Li, and Bo Li. 2024. Federated unlearning and Its privacy threats. *IEEE Netw.* 38, 2 (2024), 294–300. DOI: https://doi.org/10.1109/MNET.004.2300056

[115] Houzhe Wang, Xiaojie Zhu, Chi Chen, and Paulo Esteves-Veríssimo. 2024. Goldfish: An efficient federated unlearning framework. CoRR abs/2404.03180, (2024). DOI: https://doi.org/10.48550/ARXIV.2404.03180

[116] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. 2022. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*. 622–632.

[117] Pengfei Wang, Wei Song, Heng Qi, Changjun Zhou, Fuliang Li, Yong Wang, Peng Sun, and Qiang Zhang. 2024. Server-initiated federated unlearning to eliminate impacts of low-quality data. *IEEE Trans. Serv. Comput.* 17, 3 (2024), 1196–1211. DOI: https://doi.org/10.1109/TSC.2024.3355188

[118] Pengfei Wang, Zongzheng Wei, Heng Qi, Shaohua Wan, Yunming Xiao, Geng Sun, and Qiang Zhang. 2024. Mitigating poor data quality impact with federated unlearning for human-centric metaverse. *IEEE J. Sel. Areas Commun.* 42, 4 (2024), 832–849. DOI: https://doi.org/10.1109/JSAC.2023.3345388

[119] Pengfei Wang, Zhaohong Yan, Mohammad S. Obaidat, Zhiwei Yuan, Leyou Yang, Junxiang Zhang, Zongzheng Wei, and Qiang Zhang. 2023. Edge caching with federated unlearning for Low-latency V2X communications. *IEEE Communications Magazine* (2023), 1–7. DOI: https://doi.org/10.1109/MCOM.001.2300272

[120] Shuyi Wang, Bing Liu, and Guido Zuccon. 2024. How to forget clients in federated online learning to rank?. In *European Conference on Information Retrieval*. Springer, 105–121.

[121] Weiqi Wang, Zhiyi Tian, and Shui Yu. 2024. Machine unlearning: A comprehensive survey. CoRR abs/2405.07406, (2024). DOI: https://doi.org/10.48550/ARXIV.2405.07406

[122] Weiqi Wang, Zhiyi Tian, Chenhan Zhang, An Liu, and Shui Yu. 2023. BFU: Bayesian federated unlearning with parameter self-sharing. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 567–578.

[123] Weiqi Wang, Chenhan Zhang, Zhiyi Tian, and Shui Yu. 2024. Machine unlearning via representation forgetting with parameter self-sharing. *IEEE Trans. Inf. Forensics Secur.* 19, (2024), 1099–1111. DOI: https://doi.org/10.1109/TIFS.2023.3331239

[124] Zhen Wang, Daniyal M. Alghazzawi, Li Cheng, Gaoyang Liu, Chen Wang, Zeng Cheng, and Yang Yang. 2023. FedCSA: Boosting the convergence speed of federated unlearning under data heterogeneity. In *2023 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking (ISPA/BDCloud/SocialCom/SustainCom'23)*. IEEE, 388–393.

[125] Zichen Wang, Xiangshan Gao, Cong Wang, Peng Cheng, and Jiming Chen. 2024. Efficient vertical federated unlearning via fast retraining. *ACM Transactions on Internet Technology* 24, 2 (2024), 1–22.

[126] Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. 2022. Model extraction attacks on graph neural networks: Taxonomy and realisation. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 337–350.

[127] Bang Wu, Xingliang Yuan, Shuo Wang, Qi Li, Minhui Xue, and Shirui Pan. 2023. Securing graph neural networks in MLaaS: A comprehensive realization of query-based integrity verification. CoRR abs/2312.07870, (2023). DOI: https://doi.org/10.48550/ARXIV.2312.07870

[128] Bang Wu, He Zhang, Xiangwen Yang, Shuo Wang, Minhui Xue, Shirui Pan, and Xingliang Yuan. 2023. GraphGuard: Detecting and counteracting training data misuse in graph neural networks. CoRR abs/2312.07861, (2023). DOI: https://doi.org/10.48550/ARXIV.2312.07861

[129] Chen Wu, Sencun Zhu, and Prasenjit Mitra. 2022. Federated unlearning with knowledge distillation. CoRR abs/2201.09441, (2022). Retrieved from https://arxiv.org/abs/2201.09441

[130] Chen Wu, Sencun Zhu, and Prasenjit Mitra. 2023. Unlearning backdoor attacks in federated learning. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*.

[131] Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Jingren Zhou. 2023. On Knowledge editing in federated learning: Perspectives, challenges, and future directions. CoRR abs/2306.01431, (2023). DOI: https://doi.org/10.48550/ARXIV.2306.01431

[132] Nan Wu, Xin Yuan, Shuo Wang, Hongsheng Hu, and Minhui Xue. 2024. Cardinality counting in "Alcatraz": A privacy-aware federated learning approach. In *Proceedings of the ACM on Web Conference 2024*. 3076–3084.

[133] Hui Xia, Shuo Xu, Jiaming Pei, Rui Zhang, Zhi Yu, Weitao Zou, Lukun Wang, and Chao Liu. 2023. FedME2: Memory evaluation & erase promoting federated unlearning in DTMN. *IEEE Journal on Selected Areas in Communications* 41, 11 (2023), 3573–3588. DOI: https://doi.org/10.1109/JSAC.2023.3310049

[134] Zuobin Xiong, Wei Li, Yingshu Li, and Zhipeng Cai. 2023. Exact-Fun: An exact and efficient federated unlearning approach. In *IEEE International Conference on Data Mining (ICDM'23)*, Shanghai, China, IEEE, 1439–1444. DOI: https://doi.org/10.1109/ICDM58522.2023.00188

[135] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. 2023. Machine unlearning: A survey. *Comput. Surveys* 56, 1 (2023), 1–36.

[136] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. 2024. Machine unlearning: Solutions and challenges. *IEEE Trans. Emerg. Top. Comput. Intell.* 8, 3 (2024), 2150–2168. DOI: https://doi.org/10.1109/TETCI.2024.3379240

[137] Rui-Zhen Xu, Sheng-Yi Hong, Po-Wen Chi, and Ming-Hung Wang. 2023. A revocation key-based approach towards efficient federated unlearning. In *2023 18th Asia Joint Conference on Information Security (AsiaJCIS'23)*. IEEE, 17–24.

[138] Yang Zhao, Jiaxi Yang, Yiling Tao, Lixu Wang, Xiaoxiao Li, and Dusit Niyato. 2023. A survey of federated unlearning: A taxonomy, challenges and future directions. *arXiv preprint arXiv:2310.19218* (2023).

[139] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST'19)* 10, 2 (2019), 1–19.

[140] Andrew C. Yao. 1982. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs'82)*. IEEE, 160–164.

[141] Guanhua Ye, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2023. Heterogeneous decentralized machine unlearning with seed model distillation. CoRR abs/2308.13269, (2023). DOI: https://doi.org/10.48550/ARXIV.2308.13269

[142] Wei Yuan, Hongzhi Yin, Fangzhao Wu, Shijie Zhang, Tieke He, and Hao Wang. 2023. Federated unlearning for on-device recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 393–401.

[143] Yanli Yuan, Bingbing Wang, Chuan Zhang, Zehui Xiong, Chunhai Li, and Liehuang Zhu. 2024. Toward efficient and robust federated unlearning in iot networks. *IEEE Internet of Things Journal* 11, 12 (2024), 22081–22090. DOI: https://doi.org/10.1109/JIOT.2024.3378329

[144] He Zhang, Bang Wu, Shuo Wang, Xiangwen Yang, Minhui Xue, Shirui Pan, and Xingliang Yuan. 2023. Demystifying uneven vulnerability of link stealing attacks against graph neural networks. In *International Conference on Machine Learning*. PMLR, 41737–41752.

[145] Jian Zhang, Bowen Li, Jie Li, and Chentao Wu. 2023. SecureCut: Federated gradient boosting decision trees with efficient machine unlearning. CoRR abs/2311.13174, (2023). DOI: https://doi.org/10.48550/ARXIV.2311.13174

[146] Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. 2023. FedRecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security* (2023).

[147] Xinyu Zhang, Qingyu Liu, Zhongjie Ba, Yuan Hong, Tianhang Zheng, Feng Lin, Li Lu, and Kui Ren. 2024. Fltracer: Accurate poisoning attack provenance in federated learning. *IEEE Transactions on Information Forensics and Security* (2024).

[148] Yanjun Zhang, Guangdong Bai, Mahawaga Arachchige Pathum Chamikara, Mengyao Ma, Liyue Shen, Jingwei Wang, Surya Nepal, Minhui Xue, Long Wang, and Joseph Liu. 2023. AgrEvader: Poisoning membership inference against Byzantine-robust federated learning. In *Proceedings of the ACM Web Conference 2023*. 2371–2382.

[149] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2545–2555.

[150] Zijie Zhang, Yang Zhou, Xin Zhao, Tianshi Che, and Lingjuan Lyu. 2022. Prompt certified machine unlearning with randomized gradient smoothing and quantization. *Advances in Neural Information Processing Systems* 35 (2022), 13433–13455.

[151] Chenxu Zhao, Wei Qian, Rex Ying, and Mengdi Huai. 2023. Static and sequential malicious attacks in the context of selective forgetting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Retrieved from http://papers.nips.cc/paper%5C_files/paper/2023/hash/ed4bacc8c7ca1ee0e1d4e0ef376b7ac7-Abstract-Conference.html

[152] Yian Zhao, Pengfei Wang, Heng Qi, Jianguo Huang, Zongzheng Wei, and Qiang Zhang. 2024. Federated unlearning with momentum degradation. *IEEE Internet Things J.* 11, 5 (2024), 8860–8870. DOI: https://doi.org/10.1109/JIOT.2023.3321594

[153] Yifeng Zheng, Shangqi Lai, Yi Liu, Xingliang Yuan, Xun Yi, and Cong Wang. 2022. Aggregation service for federated learning: An efficient, secure, and more resilient realization. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2022), 988–1001.

[154] Yuyao Zhong. 2024. Federated unlearning for medical image analysis. In *Fourth Symposium on Pattern Recognition and Applications (SPRA'23)*, Vol. 13162. SPIE, 36–43.

[155] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December*

8-14, 2019, Vancouver, BC, Canada, 14747–14756. Retrieved from https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html

[156] Xiangrong Zhu, Guangyao Li, and Wei Hu. 2023. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM Web Conference 2023*. 2444–2454.

[157] Xuhan Zuo, Minghao Wang, Tianqing Zhu, Lefeng Zhang, Shui Yu, and Wanlei Zhou. 2024. Federated learning with blockchain-enhanced machine unlearning: A trustworthy approach. CoRR abs/2405.20776, (2024). DOI: https://doi.org/10.48550/ARXIV.2405.20776

Received 22 January 2024; revised 11 June 2024; accepted 15 July 2024