Abstract—Federated learning (FL) has emerged as a privacy-aware collaborative learning paradigm where participants jointly train a powerful model without sharing their private data. One desirable property for FL is the implementation of the right to be forgotten (RTBF), i.e., a leaving participant has the right to request the deletion of its private data from the global model. However, unlearning itself may not be enough to implement RTBF unless the unlearning effect can be independently verified, an important aspect that has been overlooked in the current literature. Unlearning verification is particularly challenging in FL as the unlearning effect on one participant’s data could be canceled by the contribution of other participants. In this work, we prompt the concept of verifiable federated unlearning and propose VERIFI, a unified framework that allows systematic analysis of federated unlearning and quantification of its effect, with different combinations of various unlearning and verification methods. In VERIFI, the leaving participant is granted the right to verify (RTV) to actively verify the unlearning effect in the next few rounds immediately after notifying the server of its intention to leave, along with local verification done through two steps: 1) marking that fingerprints the leaving participant by specially-designed markers and 2) checking that examines the global model’s performance change on the markers. Based on VERIFI, we have conducted so far the most systematic study on verifiable federated unlearning, covering six unlearning methods and five verification methods. Our study sheds light on the existing drawbacks and potential alternatives for both unlearning and verification methods. During the study, we also propose a more efficient and FL-friendly unlearning method “S2U”, and two more effective and robust non-invasive (without training controllability, external data, white-box model access nor introducing new security risks) verification methods “FM” and “EM”. While the proposed methods may not be a panacea for all the challenges, they address several key drawbacks of existing methods and represent a promising step toward effective, efficient, robust, and more importantly, non-invasive federated unlearning and verification. We extensively evaluate VERIFI on seven datasets, including natural/facial/medical images and audios, and four types of deep learning models, including both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). We hope, such an extensive and holistic experimental evaluation, although admittedly complex and challenging, could help establish important empirical understandings, evidence, and insights for trustworthy federated unlearning.

Index Terms—Federated learning (FL), unlearning, verification, right to be forgotten.

I. INTRODUCTION

FEDERATED learning (FL) is a collaborative learning paradigm that allows participants to jointly train a machine learning model without sharing their private data [1], [2], [3]. One required property of FL is the participants’ “right to be forgotten” (RTBF), which has been stated explicitly in the European Union General Data Protection Regulation (GDPR) [4], [5] and the California Consumer Privacy Act (CCPA) [6]. That is, a participant has the right to request the deletion of its private data. Arguably, the primary concern here is that the private data may still be memorized by the global model and continue to be exploited even after the participant leaves the federation. As leaving/joining is one frequently occurring behavior in FL, it is thus imperative to ensure that every participant can join and leave the federation freely, and more importantly, with no concerns.

The concept of machine unlearning [7], [8] has recently been proposed to remove data from a machine learning model. Several unlearning methods are designed to actively unlearn certain data from a trained model. A simple yet costly approach for unlearning is to retrain the model from scratch with the requested data being removed from the training set [7]. It can be made more efficient if the model is trained on summarized (e.g., aggregates of summations) or partitioned subsets rather than individual training samples, in which case, the model only needs to be updated on the subset(s) associated with the requested samples [9], [10]. The above methods are less practical for large-scale datasets, although advanced data partitioning strategy may help [7]. More recently, machine unlearning has been extended to FL setting, a.k.a., federated unlearning [11], [12].

1GDPR stipulates “right to be forgotten”: “The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay...”.

2CCPA specifies: “You may request that businesses delete personal information they collected from you and to tell their service providers to do the same...”.

Manuscript received 30 September 2023; revised 1 March 2024; accepted 16 March 2024. Date of publication 28 March 2024; date of current version 13 November 2024. This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62293511, Grant 62088101, Grant 62102339, and Grant 62276067, in part by the Key R&D Program of Zhejiang under Grant 2022C01018, in part by the Science and Technology Commission of Shanghai Municipality under Grant 22511016102, and in part by the National Key Research and Development Program of China under Grant 2022YFB3102100. (Corresponding author: Jiming Chen.)

Xiangshan Gao, Jingyi Wang, Shouling Ji, and Peng Cheng are with Zhejiang University, Hangzhou 310027, China (e-mail: coraja@zju.edu.cn; wangjyyee@zju.edu.cn; sjii@zju.edu.cn; lunarheart@zju.edu.cn).

Xingjun Ma is with Fudan University, Shanghai 200437, China.

Youcheng Sun is with the University of Manchester, M13 9PL Manchester, U.K.

Bo Li is with the University of Illinois Urbana-Champaign (UIUC), Champaign, IL 61820 USA.

Jiming Chen is with Zhejiang University, Hangzhou 310027, China, and also with Hangzhou Dianzi University, Hangzhou 310005, China (e-mail: cjm@zju.edu.cn).

Digital Object Identifier 10.1109/TDSC.2024.3382321
which is arguably more challenging. In FL, 1) the global model is updated based on the aggregated rather than the raw gradients; 2) FL can have a large number of participants; and 3) different participants may have similar or shared training samples\textsuperscript{3}. Consequently, gradient-based unlearning methods often rely on subtracting the reconstructed or dummy gradients of the leaving participant (a.k.a., *leaver*) to remove the associated memories, which may harm the original task or introduce new privacy risks into FL \cite{11, 12}.

Moreover, federated unlearning is only one side of the coin for the \textit{RTBF}. A more concerning question from a participant’s perspective is: \textit{how to make sure that my data has indeed been forgotten or unlearned}, which we believe is the core of establishing mutual trust in FL. Unfortunately, this important aspect has been largely overlooked in the current literature. In traditional machine unlearning, the unlearning effect can be verified by the model’s performance (e.g., accuracy and loss) on the unlearned data, additionally injected backdoor data and encryption-dependent evidence \cite{7, 9, 13, 14}. However, in FL, the accuracy and loss may hardly change when only one or a few participants left the federation, owing to the contribution of other participants. On the other hand, it is not secure in FL to use a backdoor solely for unlearning verification purposes as it could easily attack and poison the global model (see more discussion in Section V-E). Additionally, the encryption-dependent verification technique entails a considerable computational expense, which in turn limits its practical application \cite{14}. So far, it still lacks understanding of how to \textit{effectively and reliably} verify that the memories have indeed been deleted after unlearning. In fact, due to the lack of a unified, holistic and all-round FL verification framework, several key fundamental questions for trustworthy \textit{RTBF} in FL remain unexplored:

- **Federated Unlearning**: Is federated unlearning necessary or might natural forgetting be enough to forget the leaver’s data?
- **Unlearning Verification**: Do we need more sophisticated methods or simple methods like checking the global model’s performance on the leaver’s data are enough to verify unlearning?
- **Practical Choice**: What are the most effective combination(s) of unlearning and verification methods that can effectively unlearn, and clearly verify, while causing a minimal negative impact on the original task?

To answer the above questions, in this paper, we promote the concept of \textit{verifiable federated unlearning}, which treats verification as important as unlearning and grants the participant the \textit{“right to verify”} (RTV). Specifically, we design and implement \textit{VeriFi}, a unified framework for verifiable federated unlearning. The core of \textit{VeriFi} contains 1) a federated unlearning module; 2) a verification module with two key verification steps, namely \textit{marking} and \textit{checking}; and 3) a generic \textit{unlearning-verification} mechanism applicable to common FL frameworks.

\textsuperscript{3}Unlike standalone machine unlearning verification, federated unlearning verification faces the challenges of synchronous multi-leavers and different Non-IID degree among the participants, experimental discussion refers to Sections V-C and V-D.

Fig. 1 provides an overview of \textit{VeriFi}. The unlearning module can be any unlearning method adopted at the server side that erases the information of the leaving participant’s data (a.k.a., “leaving data”). The marking step of the verification module injects/tags specifically selected or designed patterns or training examples as *markers*. The checking step of the verification module then verifies unlearning based on different verification metrics defined w.r.t. the global model and the markers. The \textit{unlearning-verification} mechanism integrates all the above steps into a chained pipeline and specifies when and what to mark, and who and when to unlearn and verify.

With \textit{VeriFi}, we present a systematic, holistic, and enlightening analysis of the existing, adapted, and proposed unlearning and verification methods, shedding light on the existing drawbacks and potential alternatives.

For **Unlearning**, we study the limitations of existing one-step (e.g., differential privacy) and multi-step (e.g., retraining and gradient subtraction) unlearning methods: high cost and significant negative impact on the original task. We also propose a more efficient and FL-friendly one-step unlearning method \textit{scale-to-unlearn} (“S2U”), \textsuperscript{4}“S2U scales down the leaver’s gradients and scales up others’ gradients to encourage the global model to erase its memorization.

Verification consists of two steps: \textit{marking} and \textit{checking}. For **marking**, leveraging backdoored samples to verify the unlearning effect is popular \cite{13}, which we believe, however, is unsuitable for FL as the backdoor itself is an invasive attack that could introduce and spread new global threats across all participants. We thus propose two non-invasive unique memory-based methods that select the most representative unique individual data with the special loss pattern as the markers to verify the unlearning effect. For **checking**, we explore loss, accuracy, influence function (IF) \cite{15} and Kullback–Leibler (KL) divergence \cite{16} to comprehensively capture the performance change on the marked data (i.e., markers) caused by the unlearning.

While the proposed methods may not be a panacea for all the challenges faced by federated unlearning and verification, they address several key drawbacks of existing methods, such as the additional security risks of “BN”\textsuperscript{5} and the significant performance drop of “DP. Within the framework of \textit{VeriFi}, we conduct the first systematic study on the practicality of different combinations of unlearning and verification methods for verifiable federated unlearning. Our extensive evaluation and analysis provides answers to the three fundamental questions mentioned earlier, and establishes the empirical foundation for verifiable and trustworthy federated unlearning.

In summary, our main contributions are:

- We design the first unlearning-verification framework \textit{VeriFi} for verifiable federated unlearning. \textit{VeriFi} grants FL participants the \textit{right to verify} (RTV), i.e., the verification of the unlearning effect when leaving the federation. \textit{VeriFi} introduces a unified mechanism that allows quantitative measurement on the effectiveness of different combinations of unlearning and verification methods.

\textsuperscript{4}We use the \textsuperscript{“} symbol to indicate unlearning methods.

\textsuperscript{5}We use the \textsuperscript{*} symbol to indicate verification methods.

---

\textsuperscript{3}Unlike standalone machine unlearning verification, federated unlearning verification faces the challenges of synchronous multi-leavers and different Non-IID degree among the participants, experimental discussion refers to Sections V-C and V-D.

\textsuperscript{4}We use the \textsuperscript{“} symbol to indicate unlearning methods.

\textsuperscript{5}We use the \textsuperscript{*} symbol to indicate verification methods.
With VERIFI, we identify the limitations of existing unlearning and verification methods, and propose a more efficient and FL-friendly unlearning method $U2S$ and two more effective and robust non-invasive unique memory-based verification methods ($vFM$ and $vEM$) with the advantages empirically demonstrated. While the proposed methods may not be a panacea, they address several key shortcomings of the existing methods.

With VERIFI, we systematically study six unlearning methods and five verification methods (i.e., five marking methods and four checking metrics) with both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) on seven datasets, including three natural image, one facial image, one audio, and two medical image datasets. Our extensive and holistic experimental evaluation unveils the necessity, limitations, and potentials of different federated unlearning and verification methods, which could help inspire future work toward trustworthy federated unlearning verification.

II. PRELIMINARIES

A. Federated Learning

In FL, a number of participants jointly train a global model by communicating gradients or model parameters with a central server. At each communication round, the participants download the global model from the server, perform a certain number of local training on their private data, and then upload the accumulated local updates (gradients) to the server. The server then aggregates (e.g., using FedAvg [17]) the accumulated local updates to update the global model. The complete FL procedure is also included in Fig. 1. The participants’ private data is protected during the entire FL process, as it never leaves the local devices.

Let $[n] = \{1, \ldots, n\}$ be the set of $n$ participants with each participant owning a private dataset $D_i$ for $i \in [n]$, and $D = D_1 \cup D_2 \cup \cdots \cup D_n$ is the full training dataset. At the $t$-th communication round, the $i$-th participant first downloads the global model $w_t$, and then performs local training, e.g., using Stochastic Gradient Descent (SGD), on the local data $D_i$ to obtain an updated local model $w_t(i)$. The accumulated gradient $w_t(i) - w_t$ is then sent to the server for the global model update, e.g., using FedAvg [17] under the global learning rate $\eta$ as follows:

\[
w_{t+1} = w_t + \frac{\eta}{n} \sum_{i \in [n]} (w_t(i) - w_t).
\]  

(1)

Besides FedAvg, other aggregation rules are proposed for Byzantine-robust FL: Krum [18], Median [19], Bulyan [20] and Trimmed Mean [19]. Meanwhile, FL can be either horizontal where the participants share the same feature space but own different data samples, or vertical where the participants share the same data sample IDs but possess different features. In this work, we primarily focus on a typical horizontal FL employing FedAvg, as defined in (1). Furthermore, we specially introduce secure aggregation scheme [21] to safely update the global model without disclosure of individual updates.

B. Federated Unlearning and Verification

\textbf{Federated Unlearning:} It has been shown that deep neural networks have both memorization and forgetting effects [22], [23], [24], i.e., they naturally memorize and forget certain information about the training data. Different from natural forgetting,\footnote{The participant leaves with no active unlearning conducted by the server.} machine unlearning explicitly forces a model to forget its memorization of a target (requested to delete) subset of training samples [7]. Intuitively, unlearning can be achieved by (re)training the model on the updated dataset with the requested
samples removed. In traditional machine learning, this can be done via expensive retraining, or more efficient partition-based learning with data partition and model aggregation [7], [9], [10]. Noise can also be used to smooth out the memorization of particular samples [8]. However, in FL, information is shared via gradients. This motivates the two pioneering works [11], [12] in federated unlearning to subtract the calibrated or generated gradients of the leaver to achieve unlearning. We will incorporate the two methods in VeriFi and propose a more effective unlearning method for FL. More detailed analysis about the pros and cons of the unlearning methods in VeriFi can be found in Sections III-B and IV.

Unlearning Verification: Intuitively, the effectiveness of machine unlearning can be simply verified by the model’s performance change on the leaving data before and after unlearning [7]. Traditional encryption technique could facilitate the process of tracking unlearning, at the cost of substantial overhead [14]. Besides, unlearning can also be particularly verified on a set of backdoored training samples [13] obtained via a backdoor attack, which is essentially a data poisoning process that injects a trigger pattern into a small subset of training data so as to trick the model into memorizing the correlation between the pattern and a target class [25], [26]. Suppose the trigger pattern is \( r \) and its associated backdoor target class is \( y_{\text{target}} \). Once the trigger is learned by the model \( f \), the model will constantly predict the target class on any samples attached with the trigger pattern:

\[
\arg \max f(x \oplus r) = y_{\text{target}}, \forall (x, y) \in D, \tag{2}
\]

where, \( f \) outputs the class probabilities, the operation \( x \oplus r \) produces a backdoored version of \( x \), \( (x, y) \) is an input-label pair, and \( D \) is the training dataset in traditional machine learning. If the unlearning is effective, then the model will forget the backdoor correlation and predict the correct class instead:

\[
\arg \max \widetilde{f}(x \oplus r) = y, \forall (x, y) \in D, \tag{3}
\]

where \( \widetilde{f} \) denotes the model after unlearning and \( y \) is the correct class of \( x \).

Despite the existence of numerous unlearning methodologies, the intricacies and potential challenges of unlearning verification remain largely unexplored, particularly in the field of federated learning (FL). As the only dedicated unlearning verification work [13], pertaining solely to traditional machine unlearning, it leveraged backdoored samples to attain heightened sensitivity in verification and further raised concerns regarding security as backdoor can pose a substantial threat to all participants involved. Consequently, implementing backdoor-based verification in FL is not an optimal solution.

We also adapt and study two plausible concepts from the deep learning intellectual property (IP) protection domain for unlearning verification: watermarking [27], [28] and fingerprinting [29]. Watermarking is an invasive technique that embeds owner-specific binary string or special triggers into the model parameters to help determine the ownership of the model at a later (post-deployment) stage, while fingerprinting generates new samples to fingerprint the model’s unique properties like decision boundary [29]. In this work, we specially design and adapt these two types of techniques for federated unlearning verification. More systematic analysis of different verification methods can be found in Sections III-C and IV-C.

III. PROPOSED VERIFI FRAMEWORK

In this section, we present VeriFi framework in detail. Lying at the core of VeriFi is our proposed unlearning-verification mechanism. As illustrated in Fig. 2, the mechanism defines the timeline when unlearning and verification should be performed, and by whom, i.e., the central server or the leaver. Here, we focus on unlearning one leaver and its verification in FL in Fig. 2.

A. Unlearning-Verification Mechanism

Suppose the entire FL process consists of \( T_{\text{total}} \) communication rounds. As shown in Fig. 2, the mechanism divides the entire process into two stages, including a free stage \([T_0, T_{\text{enabled}}]\) and an unlearning-enabled stage \((T_{\text{enabled}}, T_{\text{total}})\). The free stage refers to an early FL stage where the global model has not yet converged to a good solution. In this stage, all participants can join and leave the federation freely without activating the unlearning mechanism, as in this stage, the next round of training often overwrites the model’s memorization at the previous rounds. In practice, \( T_{\text{enabled}} \) can be determined using common convergence indicators in FL, such as the gradient variance [30]. The global model changes frequently and significantly before \( T_{\text{enabled}} \), which makes natural forgetting possible. Besides, the initial learning before \( T_{\text{enabled}} \) plays a significant role in FL. Once impaired by the leaver, the degradation is irreversible [31]. Only leaving after \( T_{\text{enabled}} \) could activate the unlearning and verification process, as at this time, the model’s memorization of the private data is stabilized. Note that joining the federation at this stage should also be carefully examined as it is a harvest stage where a small contribution can receive a big reward, i.e., a high-performance global model.

Fig. 2 shows the pipelined unlearning-verification mechanism with a single leaver.\(^7\) The detailed steps can be found in Algorithm 1. In this paper, we mainly focus on one leaver per round while leaving more complex scenarios to future work. Specifically, the leaver (denoted by \( a \)) first notifies the server of its intention to leave at \( t_m \) (Step 2). Meanwhile, the leaver applies a marking method to mark the data (e.g., private training samples, triggers, or model parameters) that needs to be checked against unlearning (Step 3.a). We call the marked data ‘markers’. Once the marked model is uploaded to the server (Step 3.b), the leaver notifies the server to apply the unlearning method to unlearn its data (Step 3.c). Note that the server may or may not be aware of the existence of markers since the verification right is in the hands of the participants, not the server. The server-side unlearning may last for more than one communication round (Step 4.a). Note that, the leaver protects its true identity, using pseudonyms [32] or Tor [33], from being exposed to the central server during the checking process (Step 5.a), which would potentially mitigate the deceptive actions launched by the server. For example, the server is ignorant of when and who

\(^7\)VeriFi is easily extendable to the situation where multiple leavers require to be unlearned and verified, see Section V-C.
Algorithm 1: Federated Unlearning-Verification Mechanism.

Input: Unlearning-enabled stage $(T_{enabled}, T_{total})$, marking starting point $t_m \in (T_{enabled}, T_{total})$, unlearning starting point $t_u \in (t_m, t_{leave})$, leaving point $t_{leave} \in (t_u, T_{total})$, checking metric threshold $\delta$, marking function $\phi(\cdot)$, unlearning function $\varphi(\cdot)$, checking function $\psi(\cdot)$, aggregation rule $\text{Agg}(\cdot)$

1) Free stage: Vanilla FL before $T_{enabled}$

2) At $t_m$, participant a notifies the server to leave

3) Marking at $t_m$:
   a) a marks its local model $w_{t+1}^{(a)} \leftarrow \phi(w_{t+1}^{(a)})$
   b) a uploads the marking update $w_{t+1}^{(a)} - w_t$ to server
   c) a notifies the server the completion of marking

4) Unlearning at $t_u \in [t_m + 1, t_{leave})$:
   a) Server performs aggregation and unlearning: $w_{t+1} \leftarrow \text{Agg}(\varphi\{w_{t+1}^{(i)} - w_{t+1}^{(i)}\}_{i=1}^{n})$

5) Checking:
   a) a employs anonymous technique to evade exposing its true identity to the server
   b) a checks if unlearning is satisfied: $\psi(w_{t+1}^{(a)}) - \psi(w_{t+1}) \geq \delta$

6) Leaving at $t_{leave}$:
   a) a leaves with assured privacy if $\psi(w_{t+1}^{(a)}) - \psi(w_{t+1}) \geq \delta$
   b) a leaves with distrust if $\psi(w_{t+1}^{(a)}) - \psi(w_{t+1}) < \delta$

Fig. 2. Our proposed unlearning-verification mechanism, divided into 1) free stage, when no unlearning and verification is supported; 2) unlearning-enabled stage, which specifies the unlearning verification pipeline, including when and what to mark, who and when to unlearn and verify.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
as one of our baselines — natural forgetting "NF (more details can be found in Section IV-A). Additionally, we examine the possibility of a malicious participant who seeks to undermine the verification process by using random weights, as shown in Section V-B, which hardly work under Byzantine robust aggregation rules. Apart from these adversarial settings, we maintain certain foundational assumptions. Specifically, the local data of the involved participants remain static and the server adopts a partial device participation strategy in each round, where it randomly and equally selects a subset of the qualified participants among the involved alternatives to update the global model.

B. Unlearning

Unlearning (indicated by the subscription symbol \( ^u \)) is performed by the server immediately after marking. VERIFI includes six unlearning methods: three existing, two adapted, and one newly proposed (\(^u\)S2U), as summarized in Table I. \(^u\)RT achieves exact unlearning by rejecting the leaver’s contribution and retraining the model from scratch [10]. \(^u\)CGS, \(^u\)GGS, and \(^u\)IGS all exploit gradient subtraction to erase the leaver’s memories but with different gradient reconstruction strategies [11], [12]. \(^u\)DP achieves approximate unlearning [8], relying on the rigorous noise in differential privacy [35]. Considering the high cost and negative impact on the original task of these existing unlearning methods, we further propose \(^u\)S2U, which is more efficient, friendly, and compatible with FL.

1) Proposed Scale-to-Unlearn (\(^u\)S2U): \(^u\)S2U is inspired by the observation that scaling up/down the uploaded updates can substantially influence the global model [11], [12], [36]. Intuitively, scaling up/down one’s local update would increase/reduce its contribution to the global model. When unlearning is activated, \(^u\)S2U erases the contribution of the leaving data from the global model as follows:

\[
\varphi \left( w_{t+1}^{(j)} - w_{t} \right) = \begin{cases} 
\alpha \left( w_{t} - w_{t+1}^{(a)} \right), & \text{if } j \notin \mathcal{A} \\
\beta \left( w_{T_{\text{enabled}}} - w_{t} \right), & \text{if } j \in \mathcal{C} \setminus \mathcal{A}
\end{cases}
\]

(4)

where \( t_{\text{u}} \in (T_{\text{enabled}}, T_{\text{total}}) \) indicates the current unlearning round (see Fig. 2), \( \alpha \in [0, 1] \) is the down-scaling ratio, while \( \beta \in (1, +\infty) \) is the up-scaling ratio, and \( \mathcal{C} \) records the selected participants in FL at \( t_{\text{u}} \). Within the unlearning-enabled stage, all local models are expected to undergo minimal parameter changes, allowing \(^u\)S2U to leverage the global model at \( T_{\text{enabled}} \) to approximate other participants’ local models at \( t_{\text{u}} \):

\[
w_{t+1}^{(j)} = w_{T_{\text{enabled}}} \forall j \in \mathcal{C} \setminus \mathcal{A}
\]

Note that \(^u\)S2U does not need an accurate approximation here. By scaling up/down others’/a’s local update at \( t_{\text{u}} \), \(^u\)S2U tends to increase a’s distance to other participants’ local updates, thus actively forcing the model to unlearn a. Once a is successfully unlearned by \(^u\)S2U, the global model is brought closer to other participants’ local models while being pushed further away from a’s local model. Notably, \(^u\)S2U is compatible with the most commonly used aggregation rules such as FedAvg [17] and Krum [18].

2) Existing or Adapted Unlearning Methods: Retraining method \(^u\)RT retraining method from scratch to remove the lever’s contribution, which makes an ideal unlearning effect. Specifically, \(^u\)RT first reverts the global model to the starting point \( w_{0} \), then retraining the model from scratch without the lever’s local gradients.

Gradient subtraction methods, including calibrated gradient subtraction (\(^u\)CGS), generated gradient subtraction (\(^u\)GGS) and individual gradient subtraction (\(^u\)IGS), erase the leaver’s contribution by subtracting its contributed gradients. \(^u\)CGS [11] leverages a calibration algorithm to approximate the leaver’s gradients from other participants’ historical updates. \(^u\)GGS [12] deploys a trainable dummy gradient generator to produce such gradients that could lower the performance on the leaver’s data. \(^u\)IGS is adapted from the above two methods and directly subtracts the previously stored local updates of the leaver during the next few rounds of aggregation. Formally, gradient subtraction can be simplified as follows:

\[
\varphi \left( w_{t+1}^{(a)} - w_{t} \right) = -\lambda \sum_{i \in \Omega} \left( \tilde{w}_{t+1}^{(a)} - w_{t} \right),
\]

(5)

where \( \tilde{w}_{t+1}^{(a)} \) indicates a’s approximate local gradient (calibrated, generated, or directly stored) to be unlearned when \( i < t \) and \( i \in \Omega \), \( \Omega \) records the rounds when a was selected to contribute before \( t \), \( \lambda \) is a hyper-parameter balancing unlearning and the original task.

Differential Privacy method, \(^u\)DP [8] adds noise to a’s local updates at \( t_{u} \) to smooth out the sensitive information and blur the memorization of a’s private data in the global model:

\[
\varphi \left( w_{t+1}^{(a)} - w_{t} \right) = \epsilon \left( w_{t+1}^{(a)} - w_{t} \right) + \delta, \quad t = t_{u},
\]

(6)

where \( \epsilon \) is the privacy budget, \( \delta \) is a relaxation term, the smaller \( \epsilon \), the more noise is added. The central server introduces and adjusts (\( \epsilon, \delta \)) parameter pair to blur the global model’s memorization about a.

Discussion: Among the above six unlearning methods, \(^u\)RT achieves exact unlearning via retraining, while being very costly. The four multi-step methods (see Table I), including one retraining and three gradient subtraction methods, all need to perform unlearning for multiple communication rounds (ideally, the same number of rounds as the leaver’s previous contribution).

---

**TABLE I**

**SUMMARY OF UNLEARNING METHODS, INCLUDING EXISTING, ADAPTED, AND NEWLY PROPOSED**

| Mechanism          | Method                           | Source | Description                                                                 |
|--------------------|----------------------------------|--------|-----------------------------------------------------------------------------|
| Multi-Step         | Retraining ([RT])                | Existing | Subtract the calibrated unlearned gradients by leveraging others’ historical updates |
|                    | Generated Gradient Subtraction ([CGS]) | [11]  | Subtracted the unlearned gradients produced by a trainable dummy generator |
|                    | Individual Gradient Subtraction ([IGS]) | [7], [9]  | Subtract the leaver’s originally stored contributed gradient |
| One-Step           | Cover by noise                   | [8]  | Cover the memory by introducing noise                                       |
|                    | Differential Privacy ([DP])      | Adapted | Scale up others’ gradients and scale down the leaver’s gradients             |
|                    | Scaling ([S2U])                  | Proposed | Scale up others’ gradients and scale down the leaver’s gradients             |

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
As such, these methods need to store the calibrated, generated, and raw local/global gradients at each round, raising new privacy and gradient leakage concerns. Both 4DP and our proposed 4S2U are one-step methods that only use the current round of gradients, so are lightweight and do not need to store the local or global gradients. By involving noise in the gradients, 4DP may hurt the original task as FL heavily relies on high-quality gradients to converge. Compared with 4DP, our 4S2U is more FL-friendly as it has the minimum (or even positive) impact on the global model after aggregation, see Section IV-C.

C. Verification

Verification is performed by the leaver, consisting of two chained steps: marking and checking. Note that once a marking method is determined, so does its checking method or metrics. In Table II, we adopt, adapt or propose five marking methods for unlearning verification. 4FM and 4EM are our proposed non-invasive verification methods. 4BN inherits the backdoor-based machine unlearning verification in [13], while raising new security risks. 4ME and 4BF are both adapted from the deep learning intellectual property protection field [27], [29].

1) Marking: We call the marked information ‘markers’, a concept that is analogous to the biomarkers used in biomedical studies [37]. Intuitively, markers can be any information related to the leaver, e.g., a subset of local samples, gradients, or models. Table II summarizes the characteristics of the marking methods.

| Marking | Source | Type | Marker | Checking |
|---------|--------|------|--------|----------|
| 4FM     | -      | -    | -      | -        |
| 4EM     | -      | -    | -      | -        |
| 4BF     | -      | -    | -      | -        |
| 4ME     | -      | -    | -      | -        |
| 4BN     | -      | -    | -      | -        |

**Proposed Unique Memory Markers:** In this study, we aim to harness the unique memories of the global model concerning the leaving data as potent markers by delving into two types of unique memories: forgettable memory and erroneous memory.

**Forgettable memory (4FM)** refers to the subset of forgettable examples by the global model. Intuitively, forgettable examples are the hardest and unique examples owned by the leaver, whereas unforgettable examples are easy examples shared across different participants [38]. 4FM determines forgettable examples by the variance of their local training loss and chooses a subset of samples with the highest loss variance across several communication rounds as the markers. 4FM markers satisfy 1) hard-to-learn with high loss variance; 2) less likely to be shared with others, regarding the loss on the easy and common data would be low and steady once learned from the leaver or others. Fig. 3 illustrates a few forgettable examples (i.e., markers) identified by 4FM from MNIST [39]. We denote the marker set found by 4FM for the leaver a as $D^m_a$ and $D^m_a \subseteq D_a$. At the marking step, a locally fine-tunes the model for a sufficient number of iterations to reduce the local loss variance on $D^m_a$, then uploads the fine-tuned parameters to the server for making the global model maintain a relatively low loss variance on $D^m_a$. During checking, a can monitor the global model’s loss variance on $D^m_a$ to verify the unlearning effect. Effective unlearning should quickly recover the high loss variation on $D^m_a$ with the particularly designed low loss variance pattern caused by a’s meticulously-formed contribution is removed.

**Erroneous memory (4EM)** refers to the subset of erroneous (incorrectly predicted) samples to the global model. Intuitively, erroneous samples are likely to be the hard and rare samples uniquely owned by the leaver, as otherwise they should be well learned by the global model to form a low loss once learned from others. As described in Algorithm 2, 4EM first investigates the top $\kappa$% of the high loss samples (Line 1) and selects the majority class of erroneous samples into the marker set $D^n_a$ (Line 2). Note that the marker set has only one class (i.e., the majority class). Fig. 4 shows a few erroneous MNIST samples identified by 4EM, in which images of class ‘7’ are misclassified as ‘2’. 4EM then relabels $D^n_a$ to its mostly predicted label by the local model $f^{(2)}$ (Lines 4-6) and fine-tunes the local model on the relabelled dataset to obtain a marked model $f^{(2)}$ (Line 7). The marked model will then be uploaded to the central server to be aggregated into the global model. Fine-tuning with erroneous labels in the marking process is to make the loss on the markers smaller and check if the global model can increase the loss on the markers through unlearning. Since effective unlearning should

---

**Table II**

| Category       | Method          | Source | Type    | Marker                  | Checking |
|----------------|-----------------|--------|---------|-------------------------|----------|
| Wastewater     | Model Embedding | Adapted | Invasive | Embedded bits in model   | Matching rate of the extracted bits |
| Fingerot      | Boundary trigger | Adapted | Invasive | Boundary samples        | Accuracy on the boundary samples   |
| Unique Memory  | Forgettable Mem | Proposed | Non-invasive | Forgettable samples      | Variance of loss on the forgettable samples |
|                | Error Mem       | Proposed | Non-invasive | Error Mem samples       | Loss on the erroneous samples       |

---

**Fig. 3.** Forgettable examples (markers) found by 4FM for one leaver during 10-participant/100-alternative FL on MNIST [39].

**Fig. 4.** Erroneous samples (markers) found by 4EM for one leaver during 10-participant/100-alternative FL on MNIST. (a): a normal image from class ‘2’; (b): the erroneous images with majority true class ‘7’ but mostly predicted as ‘2’.
Algorithm 2: Erroneous Memory Marking.

Input: The local model $f^{(a)}$ and private data $D_a$ of leaver $a$, erroneous sample proportion $\kappa$, fine-tuning iterations $T^{(ft)}_f$.

Output: Marked local model $\tilde{f}^{(a)}$, marker dataset $D^m_a$.

1: $D^f_a \leftarrow \text{top } \kappa \% \text{ of high loss samples (and labels) in } D_a$
2: $D^m_a \leftarrow$ the majority class of samples in $D^f_a$
3: $\tilde{D}_a \leftarrow D_a \setminus D^m_a$
4: $\textbf{for each} (x, y) \in D^m_a \textbf{ do}$
5: $y \leftarrow \text{the most predicted label on } D^m_a$
6: $\textbf{end}$
7: $\tilde{f}^{(a)} \leftarrow$ fine-tune $f^{(a)}$ on $\tilde{D}_a \cup D^m_a$ for $T^{(ft)}_f$ iterations
8: $\textbf{return } \tilde{f}^{(a)}, D^m_a$

delete $a$'s contribution, especially the meticulously formed low loss on the "EM markers, and quickly recover the high loss on these markers.

Existing or Adapted Marking Methods: Existing watermarking methods such as parameter-based [27] and backdoor-based watermarking [28] or fingerprinting methods [29] from the field of deep learning intellectual property protection can be adapted as marking methods.

For watermarking, we adopt the backdoor-based ("BN) marking method from [13] that was initially proposed for traditional machine unlearning verification. BN leverages the BadNets [25] backdoor attack to inject the trigger patterns associated with a backdoor class into the global model to verify the unlearning effect. At the marking step, BN fine-tunes the local model on backdoored data and uploads the backdoored local parameters to the server for aggregation. After fine-tuning, backdoored samples exhibit a high attack success rate on the backdoored local and global models. Effective unlearning should break the strong correlation between the trigger pattern and the backdoor class, i.e., lowering the attack success rate on the BN markers.

For fingerprinting, we adapt the boundary fingerprint ("BF) [29] to find the decision boundary fingerprints (markers) to verify unlearning. BF generates adversarial examples that are close to the decision boundary to characterize the robustness property of the local model $f^{(a)}$. Arguably, the adversarial examples with relatively high and close top-2 class probabilities are boundary examples [29]. Therefore, before the unlearning round $t_a$ (see Fig. 2), BF marks the following adversarial examples as markers:

$$D^m_a = \{ (x, \sigma, y) \mid |f^{(a)}_{\text{top-1}}(x+\sigma) - f^{(a)}_{\text{top-2}}(x+\sigma)| \leq \gamma, (x, y) \in D_a \},$$

where, $f^{(a)}_{\text{top-1}}(x+\sigma)$ and $f^{(a)}_{\text{top-2}}(x+\sigma)$ denote the top-1 and top-2 class probabilities respectively, $x+\sigma$ is the PGD [40] adversarial example of $x$, and $\gamma \in \{0, 0.1\}$ is a small positive value defining how close are the top-2 probabilities. At the marking step, BF first fine-tunes the local model on $D^m_a$ to obtain a marked local model $\tilde{f}^{(a)}$ which now becomes robust to $D^m_a$ and has more smoothed boundary around the markers. The marked local model will then be uploaded to the central server and aggregated into the global model. Effective unlearning should quickly change the smoothed (robust) boundary around the markers (thus resulting in different predictions), which can be easily checked by the performance on the adversarial markers.

Remark: Note that certain verification methods included in this work may raise security concerns (see Section V-E), or become less effective if a secure FL aggregation rule is implemented (see Section IV-C). One example is the backdoor-based verification proposed in [13]. This aspect has also been considered when categorizing the verification techniques or making our recommendations. Therefore, we categorize the marking methods in Table II into two major types: 1) invasive methods that need to tamper with the FL process, such as modifying the global model training or injecting external data; and 2) non-invasive methods that only need to keep track of a subset of existing data. The last three columns highlight the three undesired properties: training controllability, external data and white-box access. These undesired properties may introduce new security or privacy risks into FL. Invasive methods often rely on one or more undesired properties, whereas our non-invasive unique memory-based methods avoid such limitations.

2) Checking: Checking is also performed by the leaver immediately after the marking. In this step, the change of the global model’s performance on the markers can be used to verify unlearning. This process can take a few communication rounds until the leaving time $T_{\text{leave}}$. Note that the performance is directly measured on all the leaving data if the dedicated markers in Section III-C1 are not used, as it did in most prior works [7], [9], [34].

Here, we consider four metrics (accuracy, loss, influence function [15], and KL divergence [16]) to measure the model’s performance change caused by unlearning. The accuracy and loss can be easily calculated on either the marker set or the entire leaving data. Influence function [15] formalizes the impact of a training sample on model prediction. We compute the influence function (IF) of the leave data on the global model to quantify unlearning. We refer readers to [15] for more principles and details of the influence function. KL divergence (KL) [16] measures the distributional difference between the global model’s output probability distribution and an ideal with-unlearning probability distribution $\hat{\rho}$. Arguably, the uniform distribution indicates an ideal case of unlearning, i.e., $\hat{\rho} = (\frac{1}{C}, \ldots, \frac{1}{C})$ with $C$ is the total number of classes. This gives us the following KL divergence metric checking on $D^m_a$ for unlearning verification:

$$KL(D^m_a) = E_{x \in D^m_a} \left[ f_i(x) \log \frac{f_i(x)}{\hat{\rho}} \right], \ t \in [t_m, T_{\text{leave}}].$$

The successful unlearning will lead to the uninformative prediction and a negligible KL divergence on $D^m_a$.

IV. EXPERIMENTS

We conduct extensive experiments with VerifiFi to answer the key research questions on verifiable federated unlearning defined in Section I. All experiments are conducted on a Linux server with 4 Nvidia RTX 3090 GPUs, each with 24 GB
dedicated memory, Intel Xeon processor with 16 cores and 384 GB RAM. Our code is implemented using PyTorch 1.7.1 with CUDA 11.1 and Python 3.7.

**Experimental Setup:** We run experiments on seven datasets, including two popular low-resolution image classification datasets (MNIST [39] and CIFAR-10 [41]), one speech recognition dataset (SpeechCommand [42]), two high-resolution image datasets for facial (VGGFace_mini [47]) and natural object (ImageNet_min [47]) recognition, and two medical image datasets for skin cancer (ISIC [43, 44, 45]) and COVID-19 (COVID-19) diagnoses. The datasets and corresponding models are summarized in Table III. The training data of each dataset is equally distributed to each participant, and there is no overlap between individual data. In the following part, we would explore three key and fundamental questions, including federated unlearning and verification and the practical choice, mentioned in Section I.

### A. Is Federated Unlearning Necessary?

In this study, we first examine the efficacy of Natural Forgetting (**NF**) in removing the memorization of the leaver’s data without any active unlearning. In **NF**, the server operates the same before or after the client leaves FL, without any active unlearning implemented. Specifically, we randomly select a client as the leaver and evaluate the unlearning effect of **NF** by comparing the global model’s performance on the leaving data with that obtained via Natural Training (**NT**) after all the leavers at the end of FL. The results in Table IV shows that the performance difference (the diff columns) between **NF** and **NT** are almost negligible according to all four metrics, meaning that the global model still memorizes the leaving data after **NF**. This suggests that merely rejecting the leaver’s updates without any active unlearning at the server side cannot defend against RTBF, confirming that unlearning is necessary to actively remove information about the leaver’s data.

### B. Are Markers Necessary for Verification?

In order to test whether the specialized markers are necessary for unlearning verification, we conduct experiments to verify the different unlearning effects of the six unlearning methods without using any markers (marking methods). Intuitively, if the checking metrics alone can identify the difference before and after unlearning, then the specialized markers are not necessary. For each of the four metrics (i.e., accuracy, loss, KL, and IF), we compute its normalized difference before and after (\(t_m\) and \(t_a\)) unlearning on the leaving data \(D_a\), compared with the ideal unlearning effect of **RT**. Take accuracy as an example, the normalized difference measured by the accuracy metric is computed as follows:

\[
P_{\Delta \text{Acc}} = \frac{\Delta \text{Acc} \text{, } \forall \varphi = \text{RT}}{\Delta \text{Acc}} = \frac{|\text{Acc}_{t_m}(D_a) - \text{Acc}_{t_a}(D_a)|^2}{(\text{Acc}_{t_m}(D_a))^2}.
\]

Similarly, we can define the other three metrics: \(P_{\Delta \text{Loss}}\), \(P_{\Delta \text{KL}}\), and \(P_{\Delta \text{IF}}\). We plot the normalized difference measured by the four metrics for all six unlearning methods on the seven datasets in Fig. 5. A large normalized metric difference (corresponding to the large covered area in a radar chart) indicates successful verification. For a given metric, if it successfully verifies the difference before and after unlearning across all datasets, it can be regarded as an effective metric for federated unlearning verification. Unfortunately, as shown in Fig. 5, we find that none of the metrics can effectively verify the unlearning effects of all unlearning methods, owing to other participants’ contributions. Furthermore, among the six unlearning methods, \(\text{DF and } \text{RT}\) are relatively easier to verify by any of the four metrics, as they are strong unlearning methods that either introduce a large amount of noise into the model parameters or retrain the model from scratch. Overall, our findings suggest that without specialized markers, many (4/6) of the unlearning methods may not be properly verified by the performance metrics, highlighting the imperative need for more dedicated markers to more clearly demonstrate and verify the unlearning effect.

### C. Federated Unlearning Verification With Markers

Here, we verify the unlearning effect of the six unlearning methods using the five dedicated marking methods (markers). Similarly, we compute the normalized metric\(^9\) difference of the

---

**Table III**

| Dataset                  | #classes | #samples | Resolution | Model       | Acc (%) |
|--------------------------|----------|----------|------------|-------------|---------|
| MNIST [39]               | 10       | 70,000   | 32*32      | LeNet-5     | 99.11   |
| CIFAR-10 [41]            | 10       | 60,000   | 32*32      | ResNet-18   | 95.37   |
| SpeechCommand [42]       | 10       | 46,256   | 32*32      | CNN-LSTM    | 73.09   |
| ISIC [43, 44, 45]        | 4        | 8,000    | 224*224    | DenseNet-121| 68.06   |
| COVID-19 [46]            | 3        | 16,619   | 224*224    | ResNet-18   | 88.42   |
| ImageNet_min [47]        | 10       | 13,500   | 224*224    | ResNet-18   | 90.60   |
| VGGFace_min [48]         | 20       | 7,023    | 224*224    | ResNet-18   | 95.59   |

**Table IV**

| Dataset                  | Metrics | At the Leaving Round | At the End of FL |
|--------------------------|---------|----------------------|------------------|
|                          |         | \(\text{NF}\) | \(\text{diff}\) | \(\text{NF}\) | \(\text{diff}\) |
| CIFAR-10                 | Acc (%) | 77 | 77.8 | 0.8 | 87 | 85 | 2 |
|                          | Loss    | 0.14 | 0.14 | 0.0 | 0.08 | 0.08 | 0.0 |
|                          | KL      | 6.75 | 6.92 | 0.17 | 8.31 | 8.47 | 0.16 |
|                          | IF      | 9.21e-7 | 5.67e-5 | 5.38e-5 | 1.89e-7 | 1.96e-5 |
| SpeechCommand            | Acc (%) | 64.61 | 66.67 | 0.06 | 64.73 | 67.21 | 2.48 |
|                          | Loss    | 0.50 | 0.52 | 0.02 | 0.48 | 0.5 | 0.02 |
|                          | KL      | 2.1 | 2.08 | 0.02 | 2.46 | 2.24 | 0.22 |
|                          | IF      | -0.007 | -0.006 | 0.001 | 0.006 | 0.006 | 0.0 |
| Covid                    | Acc (%) | 68.18 | 65.15 | 3.03 | 81.82 | 80.3 | 1.52 |
|                          | Loss    | 0.49 | 0.36 | 0.13 | 0.26 | 0.3 | 0.04 |
|                          | KL      | 0.31 | 0.66 | 0.35 | 1.12 | 0.99 | 0.13 |
|                          | IF      | 1.34e-5 | 0.03 | 0.03 | 0.001 | 0.001 | 0.0 |
| VGGFace_min [48]         | Acc (%) | 57.14 | 69.64 | 12.5 | 78.57 | 76.79 | 1.78 |
|                          | Loss    | 0.66 | 0.54 | 0.12 | 0.63 | 0.49 | 0.14 |
|                          | KL      | 3.03 | 3.81 | 0.78 | 3.47 | 3.99 | 0.52 |
|                          | IF      | 0.085 | 0.247 | 0.162 | 0.029 | 0.073 | 0.044 |

\(^9\)As for the principles in choosing the metrics, accuracy could provide a normalized and easily quantified unlearning effect, the **EM** and **FM** markers
global model on the marker set $D^s_a$ before and after unlearning as following:

$$R_{\Delta Mark} = \frac{\Delta Mark}{\max(\Delta Mark)}.$$

$$\Delta Mark = \frac{|Mark_{t_m}(D^s_a) - Mark_{t_m}(D^m_a)|^2}{(Mark_{t_m}(D^s_a))^2}.$$  \hspace{1cm} (10)

Due to space limitations, here we only show the most effective metric for each type of marker. As a valid marking method, apart from identifying the unlearning effect of $^u$RT and $^u$DP, which have been shown to have the better unlearning effect in Section IV-B, it should also reveal the effect of other approximate unlearning methods and distinguish the relative strengths of different unlearning methods.

**The Most Effective Verification Method: In general, our proposed $^u$EM and $^u$FM demonstrate a better verification ability than $^u$BN, $^u$ME and $^u$BF.** Particularly, $^u$EM and $^u$FM markers could always distinguish (showing larger metric differences) stronger unlearning methods ($^u$RT, $^u$DP, and $^u$S2U) from the weaker ones ($^u$CGS, $^u$IGS, and $^u$GGS) on nearly all datasets. $^u$EM performs better than $^u$FM, evidenced by the largest covered area in Fig. 6. $^u$ME (injecting a bit string into the model parameters) fails to verify $^u$DP as it is not sensitive to the noise of $^u$DP. Meanwhile, backdoor-based markers like $^u$BN fail to mark the global model at $t_m$ on the high-resolution datasets (the accuracy on $^u$BN markers of ISIC, COVID, ImageNet mini and VGGFace mini datasets is low, owing to the reduced marking effect by the aggregation rule), thus losing the reliable unlearning verification ability. $^u$BF can only verify the unlearning effect of $^u$RT and $^u$DP as other unlearning methods cause no significant change in the decision boundaries. This implies that the performance of invasive marking methods (including $^u$BN, $^u$ME, and $^u$BF) cannot guaranteed in practice.

**The Most Effective Unlearning Methods:** Upon examining the verified unlearning effect by the most effective marker $^u$EM (supported by the largest coverage area in Fig. 6), we can also cross-validate the effectiveness of the six unlearning methods. Overall, $^u$RT, $^u$DP, and our proposed $^u$S2U demonstrate a more effective unlearning effect than other unlearning methods. Note that, as a retraining method, $^u$RT arguably offers the most effective unlearning effect across nearly all datasets. The other three gradient subtraction-based unlearning methods ($^u$CGS, $^u$IGS, and $^u$GGS) exhibit limited unlearning effectiveness on all types of markers.

**Robustness to Byzantine-Robust Aggregation Rules:** We investigate the robustness of the most potent marker $^u$EM, along with two invasive markers $^u$BN and $^u$ME under the various aggregation rules. The results on the CIFAR-10 dataset in Table V clearly indicate that the metric difference identified by $^u$BN and $^u$ME drops drastically when robust aggregation rules like Krum and Median are applied at the server side.\footnote{Specifically, to avoid the instant performance change on the backdoor-based verification method, we take the median performance during $[t_m, t_m + 2]$ as the result on the markers at $t_m$.} By contrast, our $^u$EM maintains a stable difference, signifying its resilience to byzantine-robust aggregation rules. As for the reason, $^u$EM uses a meticulously selected subset of the leaving data as the markers, complying with the non-invasive nature without introducing external data, causing the marked model would not deviate so...
Fig. 6. Verifying the unlearning effect using five types of markers in (10) with each row representing one type of markers and each column representing one dataset. The six dimensions of each radar chart correspond to the six unlearning methods, with each dimension showing the normalized metric difference (and in log scale for memory-based markers) before and after unlearning. The most effective metric for markers $^v$BN, $^v$ME, and $^v$BF is accuracy, while the most effective metrics for our memory-based markers $^v$EM and $^v$FM are loss and loss variance, respectively.

TABLE V
ROBUSTNESS OF MARKERS TO DIFFERENT AGGREGATION RULES ON CIFAR-10 DATASET WITH UNLEARNING METHOD $^v$RT

| Verification | Rule   | Metrics | Before | After | diff  |
|--------------|--------|---------|--------|-------|-------|
| $^v$BN       | FedAvg | Accuracy| 72.0   | 0.0   | 72    |
|              | Krum   | Accuracy| 6.4    | 0.0   | 6.4   |
|              | Median | Accuracy| 7.0    | 0.0   | 7.0   |
| $^v$ME       | FedAvg | Accuracy| 71.88  | 48.44 | 23.44 |
|              | Krum   | Accuracy| 39.06  | 48.44 | 9.38  |
|              | Median | Accuracy| 40.62  | 48.44 | 7.82  |
| $^v$EM       | FedAvg | Loss    | 12.3   | 28.1  | 15.8  |
|              | Krum   | Loss    | 14.61  | 25.69 | 11.08 |
|              | Median | Loss    | 12.14  | 27.09 | 14.95 |

*diff*: absolute metric difference on the markers before and after vRT.

significantly from others that being excluded from contributing to the update of the global model.

Unlearning Cost: As shown in Table VI, we explore the cost of different unlearning methods, in terms of time, storage, and performance impact to the original task. Overall, our $^u$S2U demonstrates the most balanced computational overhead and impact on the original FL task, while $^u$RT requires the most time to retrain the model and $^u$CGS requires the most storage for the gradients. $^u$CGS is both time- and storage-consuming, its calibration mechanism based on others’ updates incurs the second highest time burden. By contrast, $^u$GGS and $^u$IGS have lower time and storage costs than $^u$CGS, as they directly construct the leaving gradients without using other participants’ gradients. By simply adding noise, $^u$DP requires the least time and storage cost but causes the largest performance drop. Apart from $^u$DP and $^u$RT, other unlearning methods all have a negligible impact on the global model’s performance.

Verification Cost: Here, we study the time, storage, and negative performance impact costs of different verification methods on the original FL task. As reported in Table VII, all marking methods result in a tolerable performance drop, in terms of $\Delta \text{Loss} < 0.2$ or $\Delta \text{Acc} > 3.0$. Among the five marking methods, $^v$FM has less time cost than $^v$EM and less storage burden than $^v$ME. In contrast, $^v$BN proves to be the most time-consuming method, which needs to inject the backdoor into the global model. Additionally, $^v$BF has high time and

12The minor negative impact of $^u$GGS, $^u$IGS and $^u$CGS on the original task may be attributed to the hyperparameter $\lambda$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
storage costs to save and generate the boundary fingerprints. Efficient verification is an important property for large-scale and long-term FL in real-world scenarios.

**Correlation Between the Markers and the Leaving Data:** As shown in Fig. 6, the unlearning effect is more pronounced on the markers than on the leaving data as the markers are specially designed to serve this purpose. This raises a natural question does the unlearning effect verified on the markers represent the unlearning of the full leaving data? To answer this question, we analyze the correlation between the global model’s performance on the markers and on the leaving data when adopting ‘RT as the unlearning method. As shown in Fig. 7, the performance trends on the markers and the leaving data demonstrate a positive correlation (all higher than 0.5) before and after unlearning, confirming that unlearning the markers can indeed reflect the full unlearning effect, to a large extent.

### D. Unlearning-Verification: The Combinations

The verification method goes with the unlearning method. Fig. 8 shows the normalized verifiable unlearning effect of all the combinations, composed of six unlearning methods and five verification methods on two medical datasets: ISIC and COVID. Each cell is associated with one unlearning and verification method. The darkest-colored (blue) cells highlight the best verifiable unlearning effect. Combining our analysis above, we obtain the following findings. A general effectiveness ranking of the unlearning methods is: ‘RT > ‘DP > ‘S2U > ‘CGS ≈ ‘GGS ≈ ‘IGS. Considering the high cost of ‘RT and the negative impact of ‘DP on FL, it leaves our proposed ‘S2U to be the most promising unlearning method for its relatively high effectiveness, high efficiency, high verifiability, and minor negative impact on FL. It is thus promising for future work to explore similar unlearning strategies or improve ‘DP for more effective, efficient, harmless and verifiable federated unlearning under rigorous theoretical guarantees.

As for the verification methods, the darker-colored cells are more effective. So, the general ranking is: ‘EM > ‘FM > ‘ME > ‘BF > ‘BN. For invasive verification methods, ‘BF, ‘ME, and ‘BN (i.e., the bottom three rows) only offer a limited verification effect. We rank ‘BN to the bottom of the list for the reason that backdoor should be considered as a security threat to FL (see Section V-E). Overall, our proposed ‘EM and ‘FM makers are the most promising for verification, as they are non-invasive and do not introduce external data or new security risks into FL. Meanwhile, ‘EM performs better than ‘FM with a better verification effect, supported by the larger coverage area in Fig. 6. The combination of our proposed ‘S2U-‘EM/‘FM verification appears to be the most promising unlearning-verification strategy for FL. If ‘RT and ‘DP can be somehow improved for FL, the ‘RT/‘DP-‘EM/‘FM will become the most effective combination.

### V. More Explorations

Furthermore, we investigate the unlearning-verification performance under a malicious server and participant, multi-leavers, and Non-IID data distribution, etc. By investigating these aspects, we demonstrate that the proposed unlearning-verification scheme is robust and resilient towards adversarial scenarios.

#### A. Malicious Server

We analyze an adversarial setting where the malicious server attempts to continue using the leaver’s knowledge contained in the pre-stored historical marked models, aiming for evading the unlearning verification. In fact, the capacity of the malicious server is significantly limited via the anonymous identity of the leaver during the checking process, as a result of which the malicious server can hardly determine when and who to send the unlearning-verification-targeted counterfeit model for deception. In the meanwhile, the specifically introduced secure aggregation scheme further enhances the difficulty of deceiving unlearning verification by rendering the individual updates inaccessible and providing the consistency checking against the
received global model and user sets between the leaver and other participants to prevent the server stealing the individual updates and sending the fake model to the leaver for deceiving verification. Thus, the malicious server can only stealthily use the historical global models, containing the leaver’s contribution, and hope to evade the unlearning verification. Taking the verification method *ME* as an example, which verifies unlearning based on the extracted bits from model parameters, the successfully marked model by *ME* would maintain a high and stable accuracy on *ME* markers. We assume the server implements the ideal unlearning method *RT*, then a significant accuracy decrease on the markers caused by *RT* would be observed in Fig. 9. In VeriFi, the leaver tracks the global model for a few time steps to accurately check the unlearning effect. Along with the malicious server stealthily continuing the use of the leaver’s contribution, the accuracy on the markers arises again. Such significant performance rise on the markers would be easily captured to enable the leaver immediately realizing the continued use of its knowledge. Admittedly, VeriFi provides the leaver with the relatively reliable unlearning verification result, guaranteed by the anonymous identity of the leaver in checking stage and the inaccessible individual updates and consistency check in the specifically introduced secure aggregation scheme.

### B. Malicious Participant

We explore the influence of a malicious participant, who uploads random gradients satisfying normal distribution $\mathcal{N}(\mu, \sigma^2)$ ($\mu = 0.01$ and $\sigma = 0.002$), on federated unlearning verification. We experiment with our proposed non-invasive unlearning verification methods *ME* and *FM* under the common and Byzantine robust aggregation rule: FedAvg [17] and Krum [18]. As shown in Fig. 10, *EM* and *FM* can only verify the unlearning effect of *RT* and *DP* under FedAvg in the presence of such adversarial participant. This is because the malicious participant disrupts the minimization of loss (and loss variance) on the markers at $t_m$, thereby reducing the performance gap on the markers incurred by other unlearning methods and causing failed verification under FedAvg. However, Byzantine robust aggregation rule — Krum effectively mitigates this concern by directly rejecting the contribution of the malicious participant, owing to the significant deviation between the malicious participant’s update and others’. Thus, the malicious participant cannot influence the unlearning verification effect at all, with the integration of Krum.

### C. Multi-Leavers

We also test whether our *S2U-*EM unlearning-verification combination could still provide reliable verification when multiple participants leave the federation. Here, we test two or five synchronous leavers with server-side unlearning *S2U* and client-side verification *EM*, while taking *RT* as a baseline. As Fig. 11(a) shows, under this multi-leaver setting, *EM* can still reliably verify the unlearning effect of *S2U*, especially when compared with checking the loss difference on the leaving data. This proves that our method can successfully verify the unlearning of multiple leavers, as each leaver independently chooses its own unique *EM* markers such that the unlearning effect, verified on the respective unique markers, would not interact with each other at all. However, under the constraints of computation resources, our VeriFi can only support a limited number of concurrent leavers, which may hinder its practical application to certain extent. We also take this as one limitation of our work (discussed in Section VI).

### D. Non-IID Data Distribution

We follow the Dirichlet function [49] and test the influence of Non-IID data distribution. The hyper-parameter in Dirichlet Function controls the Non-IID degree, where a smaller value represents a skewer Non-IID distribution of the local data. Here, we use 0.9 to represent mild Non-IID and 0.5 to represent extreme Non-IID. The comparison result between our *S2U* and *RT* verified by *EM* is illustrated in Fig. 11(b). It shows that, under extreme Non-IID (i.e., the participants own more


F. Unlearning Verification Visualization

Here, we visually inspect the efficacy of unlearning and verification by utilizing the interpretability algorithm, Grad-CAM [52], as depicted in Table VIII. These saliency maps of the "BN markers $x \oplus r$, patched with a 5*5 white square trigger and specified target class "1", are computed based on the global model. Before marking ($t < t_m$), the global model outputs the original class on the "BN markers and pays more attention to the key object or area. After marking and before unlearning ($t \in [t_m, t_u]$), the memorization of a still exists as the backdoor sample is classified as the target class and the high attention area is mainly on the trigger. After unlearning ($t \geq t_u$), "RT, "DP, and "S2U all exhibit a clear unlearning effect since the attention on the trigger drops significantly. The weakened attention can be owed to the gradually eliminated memory about the backdoor markers. The attention decrease on the trigger can also be observed in "CGS, "GGS, and "IGS, however, not as obvious as in other unlearning methods. With the verification method, the unlearning effect can be explicitly visualized and explained.

VI. LIMITATION

Here, we discuss several limitations of VeriFi.

- **Unlearning and verifying evolving data:** Currently, VeriFi can only operate in a static data setting and cannot support a dynamic environment where the data distribution evolves over time. The dynamic data setting poses a significant challenge as the leaving data can also be dynamic. For instance, in "GGS (unlearning by generated gradient subtraction), it becomes increasingly challenging to generate the gradients to continuously reduce the memorization of the leaver’s evolving data. As for verification, taking "EM as an example, it is more difficult to select reliable erroneous markers from the dynamic leaving data. Combining Trusted Execution Environment (TEE) with certain cryptographic techniques (hashing) may help mitigate this concern and more dedicated research may be necessary to develop more robust solutions for unlearning and verification in dynamic data environments.

- **Unlimited number of leavers:** Our VeriFi only experimented with a single leaver, although the framework also supports multiple leavers as analyzed in Section V-C. However, VeriFi is inherently limited to a low number of leavers.

| Dataset   | "BN Marker | $t < t_m$ | $t_m \leq t < t_u$ | $t \geq t_u$ |
|-----------|------------|-----------|--------------------|-------------|
| CIFAR-10  | $x \oplus r$ | ($x \oplus r, y$) | ($x \oplus r, y_{\text{target}}$) | ($x \oplus r, y_{\text{target}}$) |
| COVID     |            |           |                    |             |

(a) Orig class: 0 1  (b) Target class: horse  (c) Orig class: automobile  (d) Target class: automobile

Fig. 12. Example of "BN markers, shown in 12(b) and 12(d), once patched with the trigger, would be misclassified into the target class ‘1’ (‘automobile’ in CIFAR-10).

- **EM markers can reveal the unlearning effect of "S2U more clearly.** This is in line with our expectation as extreme Non-IID tends to produce more unique markers at each client.

- **Security Risk of "BN**

Apart from working as a watermark to verify unlearning, "BN itself is a traditional backdoor attack widely studied in the literature [36], [50], [51]. The left table in Fig. 13 shows that, even at the end of FL, the backdoor attack (samples patched with the trigger would be misclassified into the backdoor target class, as shown in Fig. 12(b) and (d)) still exists on datasets MNIST, CIFAR-10, and SpeechCommand. Not all unlearning methods can completely remove the security risk caused by such an invasive marking method. Specifically, approximate unlearning methods ("CGS, "GGS, and "IGS) cannot significantly reduce the attack success rate, even higher than 70%. Fortunately, the backdoor can be effectively removed by robust aggregation rules, such as Krum and Median, as shown in the right figure in Fig. 13.

| Dataset   | "NF" | "CGS" | "GGS" | "IGS" |
|-----------|------|-------|-------|-------|
| MNIST        | 99.57 | 99.17 | 98.67 | 98.67 |
| CIFAR-10     | 99.34 | 98.05 | 98.05 | 96.53 |
| SpeechCommand| 72.1   | 73.6  | 71.4  | 72.8  |

Fig. 13. Left table demonstrates the security threat caused by "BN, indicated by the attack success rate on the backdoor markers, which cannot be completely removed by the approximate unlearning methods ("CGS, "GGS, and "IGS). The right figure shows the reduced attack success rate by byzantine robust aggregation rules.
which may hinder the application of VeriFi in large-scale FL with thousands or millions of clients (and leavers). We hope this challenging scenario can be addressed in the future with the sufficient computation resource and our modularized and flexible mechanism design in VeriFi.

- **Adversarial robust verification:** Designing an unlearning verification method that’s both highly effective and resilient against adversarial attacks is an exceptionally challenging task. This is because FL is a rather open environment which naturally fosters a wide attacking surface, such as the malicious server and participant crafting models to deceive verification. Even though the integration of these additionally introduced measures — secure aggregation scheme, anonymous technique and Byzantine aggregation rule could mitigate the concern by rendering the individual updates inaccessible, making the leaver-targeted fake model unreachable and excluding the malicious updates by specialized selection strategy, we still expect developing such verification method that can withstand adversarial challenges without heavily relying on any additional defensive scheme, involving a deeper exploration of the leaver’s unique memory and characteristics.

- **Fuse of multi-verification:** As shown in Section IV-C, each verification method has its own characteristics. Combining several verification methods, adhering to different mechanisms, may realize the better unlearning verification effect. This is technically realizable in the pipeline of our VeriFi as it supports parallel unlearning verification using different methods.

- **Extension to other domains:** VeriFi has only explored unlearning and verification in image and speech domain, however, **RTBF** is equally important for domains including natural language processing [53], [54], reinforcement learning [55], [56], etc. We plan to extend VeriFi to support more domains in its future releases.

### VII. RELATED WORK

**Unlearning: RTBF** brings an urgent demand for certain feasible and controllable federated unlearning methods. Nevertheless, only [11] and [12] have focused on federated unlearning by subtracting the reconstructed unlearned models to safeguard **RTBF**. Liu et al. [11] leveraged the historical model updates of other participants to reconstruct and calibrate the unlearned model, violating the invisibility of the aggregator to the model and causing privacy threats by collecting others’ local model updates. Liu et al. [12] proposed to deploy a trainable dummy gradient generator to produce the submitted model updates, which could lower the performance on the leaving data, requiring extra time to train the generator and may impair the initial task. Both [11] and [12] proposed to erase the memory about the leaver by subtracting the model updates in FL, we also incorporate the two active unlearning methods in VeriFi. There are still several related works in the field of machine unlearning. Cao et al. [9] first used “machine unlearning” term and realized it by subtracting the specific transformation from the summations of the training data’s transformations in statistical query learning. Ginart et al. [10] designed an effective unlearning technique against k-means. The two unlearning methods are effective against simple machine learning algorithms, inapplicable to FL. Shintre et al. [8] discussed the possible unlearning methods briefly, such as differential privacy [35]. He et al. [57] introduced the breakpoint strategy which sacrifices storage to save the intermediate model for unlearning efficiency. Bourtoule et al. [7] proposed SISA which splits data into shards and slices, and only reetrains on the specific shard or slice, containing the forgotten data. Although effective, these methods are not suitable for FL since the forgotten data is not accessible by the server, and only the model can be directly processed to remove memories. Pascoal et al. [58] used a Trusted Execution Environment to realize the dynamic privacy-preserving federated genomic statistics platform, enabling the participants dynamically retract their data. However, only physically deleting data cannot comply with **RTBF** without deleting the related memories. Therefore, the development of federated unlearning methods that can directly operate on the uploaded model updates is necessary.

**Unlearning Verification:** Unlearning verification could establish such trust on “my data has indeed been forgotten”. Prior work on machine unlearning verification have typically relied on accuracy on forgotten data as evidence of unlearning, which is not applicable in FL, since the accuracy on the forgotten data would not drop significantly owing to others’ contributions. The traditional encryption-based verification method, which tracks data updates throughout the entire learning process, is typically limited to simple numerical datasets, owing to its complex procedures and substantial computational costs [14]. Alternative approaches such as measuring minor performance differences (including accuracy and loss) [11] or evaluating on the specially introduced backdoor data [13] have been proposed to verify whether the memory is deleted as promised. In VeriFi, we have considered the minor performance change on the metrics for checking in Sections III-C2 and IV-B, and incorporated the backdoor-based unlearning verification method in BN and disclosed the hidden security risk to all the participants under the surface of effective machine unlearning verification tool. Leveraging certain mature membership inference methods [59], [60], [61] to directly reveal the unlearning result is a promising research direction that warrants dedicated attention, while being in parallel to the main approach of verification through marking in this work.

### VIII. CONCLUSION AND FUTURE WORK

In this paper, we design and implement the first comprehensive platform — VeriFi, a unified federated unlearning and verification framework that allows systematic analysis of the verifiable unlearning with different combinations of unlearning and verification methods. Based on VeriFi, we conduct the first systematic study in the literature for verifiable federated unlearning, with six unlearning methods (including newly proposed $S2U$ and five verification methods (including newly proposed FM and EM), covering existing, adapted and newly proposed algorithms for both unlearning and verification. Extensive experiments showed that our proposed $S2U$ is an effective,
efficient, and secure federated unlearning method with low time cost, storage cost, and negative impact on the original FL task. The experiments also confirm the effectiveness of our proposed non-invasive $^{15}$FM and $^{14}$EM markers for unlearning verification. The combination of $^{14}$S2U,$^{15}$EM/$^{14}$FM yields so far the most promising results for verifiable federated unlearning without tempering with the FL process, white-box model access, or raising new security concerns. While the proposed methods may not be a panacea for all the challenges faced by federated unlearning and verification, it addresses many deficiencies of existing works (such as the security risk of $^{14}$BN and the negative performance impact of $^{14}$DP) and represents a promising step toward trustworthy federated unlearning verification.

Research on verifiable federated unlearning is emerging and our proposed VEFIT can serve as a holistic test bed for developing and benchmarking future federated unlearning and verification techniques. Following VEFIT, there are a rich set of research opportunities to explore further, such as new unlearning and verification methods, certification of federated unlearning, etc.

**REFERENCES**

[1] K. Bonawitz et al., “Towards federated learning at scale: System design,” in *Proc. Mach. Learn. Syst.*, vol. 1, pp. 374–388, 2019.

[2] P. Kairouz et al., “Advances and open problems in federated learning,” 2019, arXiv:1912.04977.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[4] GDPR, “Right to erasure (right to be forgotten),” 2017. [Online]. Available: https://gdpr-info.eu/art-17-gdpr/

[5] P. Voigt and A. Von dem Bussche, “The EU general data protection regulation (GDPR): A practical guide,” vol. 10, no. 3152676, pp. 10–5555, 2017.

[6] E. L. Harding, J. J. Vanto, R. Clark, L. Hannah Ji, and S. C. Ainsworth, “Understanding the scope and impact of the California consumer privacy act of 2018,” *J. Data Protection Privacy*, vol. 2, no. 3, pp. 234–253, 2019.

[7] L. Bourtoule et al., “Machine unlearning,” in *Proc. IEEE Symp. Secur. Privacy*, 2021, pp. 141–159.

[8] S. Shintre, K. A. Roundy, and J. Dhaliwal, “Making machine learning forget,” in *Proc. 7th Annu. Privacy Forum*, Springer, 2019, pp. 72–83.

[9] Y. Cao and J. Yang, “Towards making systems forget with machine learning: Towards optimal statistical rates,” in *Proc. Int. Conf. Mach. Learn.*, pp. 107–115.

[10] D. Yiu, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5630–5639.

[11] R. Guerraoui et al., “The hidden vulnerability of distributed learning in heterogeneous networks,” in *Proc. 7th Annu. Privacy Forum*, 2018, pp. 3521–3530.

[12] K. Bonawitz et al., “Practical secure aggregation for privacy-preserving machine learning,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[13] D. Arpit et al., “A closer look at memorization in deep networks,” in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 233–242.

[14] T. Kirkpatrick et al., “Overcoming catastrophic forgetting in neural networks,” in *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[15] G. Liu, X. Ma, C. Wang, and J. Liu, “Federated unlearning: Towards optimal statistical rates,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 159–172.

[16] X. Cao, J. Jia, and N. Z. Gong, “IPGuard: Protecting the intellectual property of deep neural networks via fingerprinting the classification boundary,” 2019, arXiv:1910.12903.

[17] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on Non-IID data,” 2019, arXiv:1907.02189.

[18] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Sato, “Embedding watermarks into deep neural networks,” in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2017, pp. 267–277.

[19] J. Zhang et al., “Protecting intellectual property of deep neural networks with watermarking,” in *Proc. Asia Conf. Comput. Commun. Secur.*, 2018, pp. 3518–3531.

[20] X. Cao, J. Jia, and N. Z. Gong, “IPGuard: Protecting the intellectual property of deep neural networks via fingerprinting the classification boundary,” 2019, arXiv:1910.12903.

[21] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, “On the convergence of FedAvg on Non-IID data,” 2019, arXiv:1907.02189.

[22] Y. Cao and J. Yang, “Towards making systems forget with machine learning: Towards optimal statistical rates,” in *Proc. Int. Conf. Mach. Learn.*, pp. 107–115.

[23] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[24] R. Dingledine et al., “Tor: The second-generation onion router,” in *Proc. USENIX Secur. Symp.*, 2004, pp. 303–320.

[25] G. Liu, Y. Yang, X. Ma, C. Wang, and J. Liu, “Federated unlearning,” 2020, arXiv:2012.13891.

[26] M. Abadi et al., “Deep learning with differential privacy,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[27] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in *Proc. Int. Conf. Artif. Intell. Inf. Process. Syst.*, 2019, pp. 3518–3531.

[28] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, “Making AI forget you: Data deletion in machine learning,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–10.

[29] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, “FedEraser: Enabling efficient client-level data removal from federated learning models,” in *Proc. IEEE/ACM 29th Int. Symp. Qual. Serv.*, 2021, pp. 1–10.

[30] Y. Liu, Z. Ma, X. Liu, and J. Ma, “Learn to forget: User-level memorization elimination in federated learning,” 2020, arXiv:2003.10933.

[31] D. M. Sommer, L. Song, S. Wagh, and P. Mittal, “Athena: Probabilistic verification of machine learning,” in *Proc. Privacy Enhancing Technol.*, vol. 3, pp. 268–290, 2022.

[32] T. Eisenhofer, D. Riepel, V. Chandrasekaran, E. Ghosh, O. Ohrimenko, and N. Papernot, “Verifiable and provably secure machine unlearning,” in *Proc. Asia Conf. Comput. Commun. Secur.*, vol. 10, no. 2, pp. 1–19, 2020.

[33] T. P. Koh and P. Liang, “Understanding black-box predictions via influence functions,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.

[34] J. Goldberger et al., “An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures,” in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 487–493.

[35] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[36] P. Blanchard et al., “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 119–129.
M. Thomas, “Estimating a Dirichlet distribution,” MIT, Tech. Rep., 2000.
Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, “Can you really backdoor federated learning?”, 2019, arXiv:1911.07963.
H. Wang et al., “Attack of the tails: Yes, you can really backdoor federated learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, Art. no. 1348.
R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 618–626.
J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv: 1810.04805.
E. Cambria and B. White, “Jumping NLP curves: A review of natural language processing research,” IEEE Comput. Intell. Mag., vol. 9, no. 2, pp. 48–57, May 2014.
R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.
L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” J. Artif. Intell. Res., vol. 4, pp. 237–285, 1996.
Y. He, G. Meng, K. Chen, J. He, and X. Hu, “DeepObliviate: A powerful charm for erasing data residual memory in deep neural networks,” 2021, arXiv:2105.06209.
T. A. Pascoal, J. Decouchant, A. Boutet, and P. J. E. Verissimo, “DyPS: Dynamic, private and secure GWAS,” in Proc. Privacy Enhancing Technol., vol. 2021, no. 2, pp. 214–234, 2021.
F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction APIs,” in Proc. 25th USENIX Secur. Symp., 2016, pp. 601–618.
R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in Proc. IEEE Symp. Secur. Privacy, 2017, pp. 3–18.
M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in Proc, 22nd ACM SIGSAC Conf. Comput. Commun. Secur., 2015, pp. 1322–1333.

Xiangshan Gao (Student Member, IEEE) received the bachelor’s degree in computer science from Xidian University, Xi’an, China, in 2018. She is currently working toward the PhD degree with the State Key Laboratory of Industrial Control Technology, Group of Networked Sensing and Control (IPC-NeSC), Zhejiang University, Hangzhou, China. Her research interests include trustworthy machine learning and cyber physical system security.

Xingjun Ma (Member, IEEE) is an associate professor with the School of Computer Science, Fudan University. His main research area is trustworthy machine learning, aiming to design secure, robust, explainable, privacy-preserving, and fair machine learning algorithms and models for AI applications. He has published at top-tier conferences and journals, including ICLR, NeurIPS, ICML, CVPR, and the IEEE Security & Privacy. He also serves as a PC member or reviewer for a number of conferences and journals.

Jingyi Wang (Member, IEEE) received the bachelor’s degree in information engineering from Xi’an Jiaotong University, in 2013, and the PhD degree from the Singapore University of Technology and Design, in 2018. He is currently a tenure-track assistant professor with the College of Control Science and Engineering, Zhejiang University, China. He was a research fellow with the School of Computing, National University of Singapore during 2019-2020 and at Information Systems Technology and Design Pillar, Singapore University of Technology and Design during 2018-2019. His research interests include formal methods, software.

Jinming Chen (Fellow, IEEE) received the PhD degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2005. He is a professor with the Department of Control Science and Engineering, Zhejiang University, and president of Hangzhou Dianzi University. His research interests include IoT, networked control, wireless networks. He serves on the editorial boards of multiple IEEE Transactions, and the general co-chairs for multiple IEEE/ACM conferences. He was a recipient of the 7th IEEE ComSoc Asia/Pacific Outstanding Paper Award, the JSPS Invitation Fellowship, and Newton Advanced Fellowships. He is an IEEE VTS distinguished lecturer and a fellow of the CAA.

Youcheng Sun (Member, IEEE) received the PhD degree from Scuola Superiore Sant’Anna in Pisa, Italy. He is an assistant professor from the University of Manchester, U.K. Prior to Manchester, he worked with Queen’s University Belfast and he was a postdoctoral researcher with the University of Oxford. A recognised expert in the field of Trustworthy AI, he has led the development of innovative techniques focused on ensuring the safety of AI systems. His research has been funded by leading companies such as Google, Ethereum Foundation, the U.K.’s Defence Science and Technology Laboratory (Dstl), and BAЕ Systems. He has also been awarded grants from The Alan Turing Institute for the U.K.-Italy Trustworthy AI visiting researcher programme.

Bo Li (Member, IEEE) is an assistant professor with the Department of Computer Science, University of Illinois at Urbana-Champaign, and is a recipient of the Symantec Research Labs Fellowship, Rising Stars, MIT TR 35, and best paper awards in several machine learning and security conferences. Previously, she was a postdoctoral researcher with UC Berkeley. Her research focuses on both theoretical and practical aspects of security, machine learning, privacy, game theory, and adversarial machine learning.

Shouling Ji (Member, IEEE) received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology, and the PhD degree in computer science from Georgia State University. He is a Qiushi distinguished professor with the College of Computer Science and Technology, Zhejiang University. His current research interests include AI security, data-driven security and privacy and software and system security. He is a member of the ACM, and a senior member of the CCF. He was a Research Intern with the IBM T. J. Watson Research Center. He is the recipient of the 2012 Chinese Government Award for Outstanding Self-Financed Students Abroad and ten Best/Outstanding Paper Awards, including ACM CCS 2021.

Peng Cheng (Member, IEEE) received the BSc and PhD degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively. He is currently executive dean, and Changjiang chair professor with the College of Control Science and Engineering, Zhejiang University. He serves as associate editors of IEEE Transactions on Control of Network Systems, IEEE Transactions on Cloud Computing. He also served as guest editors of IEEE Transactions on Automatic Control and IEEE Transactions on Signal and Information Processing over Networks. His research interests include control system security, cyber-physical systems, and cloud networking.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.