Federated Unlearning and Its Privacy Threats

Fei Wang, Baochun Li, and Bo Li

Abstract

Federated unlearning has emerged very recently as an attempt to realize "the right to be forgotten" in the context of federated learning. While the current literature is making efforts on designing efficient retraining or approximate unlearning approaches, they ignore the information leakage risks brought by the discrepancy between the models before and after unlearning. In this paper, we perform a comprehensive review of prior studies on federated unlearning and privacy leakage from model updating. We propose new taxonomies to categorize and summarize the state-of-the-art federated unlearning algorithms. We present our findings on the inherent vulnerability to inference attacks of the federated unlearning paradigm and summarize defense techniques with the potential of preventing information leakage. Finally, we suggest a privacy preserving federated unlearning framework with promising research directions to facilitate further studies as future work.

Introduction

"The right to be forgotten" in recent privacy legislation, such as the GDPR, grants users a right to request their private data be deleted. First introduced in the literature as *machine unlearning* problems, solutions were proposed to allow trained machine learning models to forget the data to be removed [1], [2]. In contrast to most existing studies that focused on centralized machine unlearning where the model owner has access to all the data, a new line of research, called *federated unlearning*, has emerged with an objective of extending the investigation of data removal and unlearning to the federated learning context. In federated learning (FL), multiple devices collaboratively train a shared model without transmitting their private data, and may join or leave the training process at any time. Erasing a client's entire or part of data from the global model can help improve the flexibility and reliability of the FL systems.

Inherited from conventional machine unlearning, *federated unlearning* was proposed to meet the "right to be forgotten" requirement, but in a distributed setting. Federated learning adds some distinctive challenges to designing an effective unlearning algorithm, which have been identified by some prior work [3], [4]. New mechanisms have recently been designed [3], [4], [5], [6] for machine unlearning in the federated learning setting, generally referred to as *federated unlearning*.

Existing work on federated unlearning usually assumes that all the data to be removed belongs to one client [3], [5] and thus the goal of the unlearning process is to erase the historical contributions of that particular client to the global model training. An intuitive, yet naïve, way to perform unlearning is to retrain the model from scratch after removing the data requested to be deleted. However, it is quite computationally expensive to do so, and it is not practical to have the same set of clients participating in the retraining process again, which was assumed by some existing rapid retraining mechanisms [4]. Rather than retraining from scratch, we believe that the only feasible way in practice to perform federated unlearning is to use approximation algorithms.

There are, however, several unique challenges in federated learning that make it unlikely to apply existing approximation algorithms in conventional machine unlearning to the federated unlearning setting. In federated learning, individual contributions from one client in each communication round will be quickly spread across the other clients, due to global aggregation at the server in subsequent rounds. As such, these contributions are difficult to be isolated and removed. In addition, in federated learning, a client always keeps its own dataset privately, restricting access from either the server or the other clients participating in the same training session. Therefore, approximation algorithms in machine unlearning, such as dataset splitting or partitioning [2], cannot be readily applied.

However, existing studies on federated unlearning only focused on the efficiency of the unlearning process, but largely overlooked its inherent vulnerability, which diminished the expected benefits of preserving privacy with federated learning. The capability of identifying if a data sample has been used for training a model, or that of reconstructing data samples from the trained model or even gradients, have been demonstrated by existing attacks such as membership inference attacks [7], [8], model inversion attacks [9], and gradient leakage attacks [10]. The unlearning framework can bring benefits to these attackers and violate the intention of erasing a client's private data without trace. From the perspective of a honest-but-curious server or an adversarial client, by taking advantage of the discrepancy between two versions of the global

Fei Wang (corresponding author) and Baochun Li are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada. Bo Li is with the Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong.

Digital Object Identifier: 10.1109/MNET.004.2300056

0890-8044/23/$25.00 © 2023 IEEE

model, *i.e.*, models before and after the unlearning process, they are able to extract private data from these two model versions. This defeats the purpose of federated unlearning, where such private data are supposed to be deleted and protected from further exposure.

In this paper, we provide a comprehensive review of the latest advances in federated unlearning, including its privacy risks that can compromise the goal of protecting clients' private data by erasing, with particular attention to membership inference attacks, as well as potential defense mechanisms. Throughout this paper, we will present original insights and commentary that have not been previously considered or emphasized in the literature.

The Emergence of Federated Unlearning

FedEraser [3] was proposed as the first attempt to approximate unlearning in the context of federated learning. It used a calibration technique to separate the individual contributions of clients as much as possible. The prerequisite is that the server has to store the history of parameter updates for every client. While this assumption is not unreasonable in practice, if there are more than a few hundred clients participating in an FL session, it may take up a large amount of storage at the server. FedEraser is simply a retraining method that relies on extra rounds of communication between the server and clients where all the client participants adjust their historical updates with their historical dataset.

Similarly, Wu et al. [5] also required the server to store the history of updates for every client. However, instead of asking clients to retrain the model as FedEraser did, Wu et al.'s [5] method only asked the server to subtract all the historically averaged updates from the target client from the final global model to get a skewed unlearning model and then use knowledge distillation to train such a skewed unlearning model, using the original global model as the teacher model on an outsourced unlabelled dataset. This method needed to sample synthetic data with the same distribution of the entire dataset, and the accuracy of such a sampling process can be negatively affected by non-IID (not independent and identically distributed) data distribution, which is typically assumed in federated learning.

Liu et al. [4] proposed a rapid retraining method that retrained the global model on the remaining dataset by approximating the loss function using the first-order Taylor expansion, which relied on the participation of all the clients. This algorithm could simply be regarded as a rapid local training algorithm, not necessarily an approximation algorithm for federated unlearning.

Halimi et al. [6] did not require the server to store parameter updates of clients and only relied on the target client who wished to opt out. The target client performed projected gradient ascent to train the global model to maximize the empirical loss on its local data before the deletion. The average of the remaining clients' models is used as a reference model to measure the quality of unlearning.

Assumptions in Existing Work. Overall, the topic of federated unlearning has not been extensively investigated yet. None of the existing studies has compared the performance of the model after unlearning with each other, except for the model after naïve retraining from scratch. As a result, it is not clear which existing algorithm performed the best with respect to the wall-clock time consumed for unlearning, or the impact on global model accuracy. Moreover, existing work used surprisingly different assumptions on the unlearning scenarios, including different unlearning targets and unlearning performers. These diverging assumptions intrinsically determine the limitations of their algorithms. In what follows, we present a detailed account of these assumptions in the existing work.

Data to be Unlearned. Existing work usually assumed that unlearning happens when a client completely opts out of the current FL session. However, there are many cases in federated learning where only a portion of a client's data is requested to be removed. Such a difference is illustrated visually in Fig. 1. Existing work that only considered one of these scenarios may fail in the other. For example, the unlearning mechanism in Wu et al. [5] will not work as expected when the target client only requests to remove a portion of its data, since the server will remove all its historical average updates from the global model.

Different clients may have similar, or to some extent, shared training samples. In this case, removing (all or part of the) data of a client from the global model will also affect the performance of the unlearned model on the remaining data of other clients. Training the global model to maximize the empirical loss on the target client's local data as Halimi et al. [6] did will naturally lead to a high loss on the same data at the other clients.

The Unlearner. Having different roles in federated learning — the server, the target client, or the remaining clients — to perform the unlearning leads to different benefits and deficiencies, since these roles have very different capabilities, access to data, and privacy leakage risks. As shown in Table 1, we categorize the existing work according

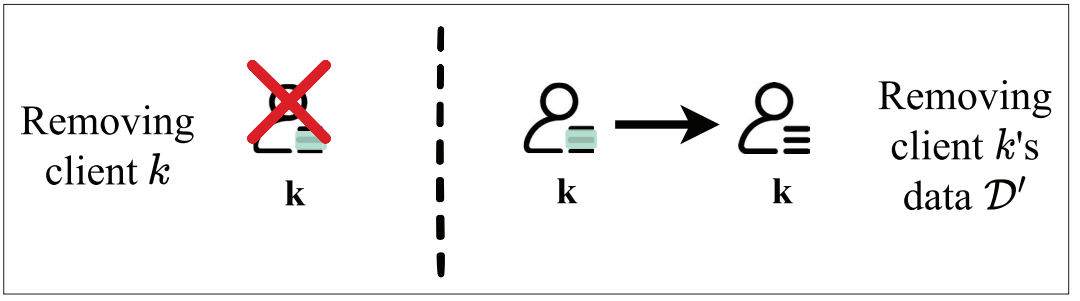

FIGURE 1. Different assumptions on data to be unlearned.

Unlearner	Pros	Cons
Server [5]	Usually has more computation and storage power than clients	Needs to store the history of every participant's parameter updates, and does not work well if only a portion of the data is requested to be removed from the target client
Remaining clients [3]	Has access to more data	Induces extra computation or communication on non-target clients
All clients [4]		
Target client [6]	Has access to the data to be deleted; can ensure the unlearning performance locally	Has limited approximation

TABLE 1. The pros and cons of different federated unlearners.

to the role that carries out the unlearning process, called the *unlearner* in this paper, and summarize the pros and cons with each role. Figs. 2a to 2c give an overview of the unlearning process under scenarios of different unlearners. We analyze each role as the unlearner as follows.

The Server (Fig. 2a). When a client requests to opt out of the ongoing FL session, the server is responsible for erasing the target client's training data from the current global model before continuing the session, and to send it out to the remaining clients in the next communication round. To do this, the server has to store the history of updates and the check-in round for each client, in order to rebuild the global model as proposed in [5]. Some papers argued that the server is competent to perform the unlearning task because it has more computation power or storage capacity than the clients. In addition, no additional rounds of communication between the server and the clients are needed for unlearning purposes. One implied limitation is that it cannot accommodate the second unlearning target, where only parts of local data is requested to be removed from a client. This is due to the fact that the server cannot distinguish the correlation between those local updates and the corresponding training data of the target client when rebuilding the global model.

The Remaining Clients or All Clients (Fig. 2b). The server may need to first roll back the global model to a previous checkpoint right before the target client was chosen for the first time. All the clients are then asked to participate in the retraining process, similar to regular FL with local fine-tuning or calibration if necessary [3], [4]. The target client may opt out prior to the retraining process commences, or may participate in the retraining process with the remaining data, for the corresponding unlearning target. This scheme requires the most extra participation and communication in the unlearning process, but it can make full use of the data owned by all the clients for retraining. In practice, however, it is not feasible to ask all the clients to help unlearn the target client or a portion of data belonging To the target client. In addition, the retraining process with calibration [3] or rapid retraining performed for unlearning purposes [4] may simply generate an offset new global model due to the stochastic characteristics of federated learning.

The Target Client (Fig. 2c). The unlearning only takes place at the target client, who has direct access to the data that needs to be deleted. After unlearning, the target client then sends the local update to the server for aggregation. This scheme is much more economical — with respect to both time and communication — than its alternatives. It is also more flexible for the target client to verify the unlearning performance locally. However, the model erasing effect of local unlearning certainly cannot compete with that of global retraining, where the model does not see the data supposed to be deleted at all. In some cases, certain clients may be less suitable for performing the unlearning process compared to others. For instance, if a client requires erasure after only a few rounds of training, it may be more practical to allow all clients to retrain from a previous checkpoint instead.

Privacy Threats to Federated Unlearning

Federated unlearning can add more risks to privacy attacks in federated learning, and more surprisingly, fail to protect the private data meant to be erased in the first place. We focus on studying the effectiveness of one major category of inference attacks — the *Membership Inference (MI)* attack and its enhanced variants — in federated unlearning.

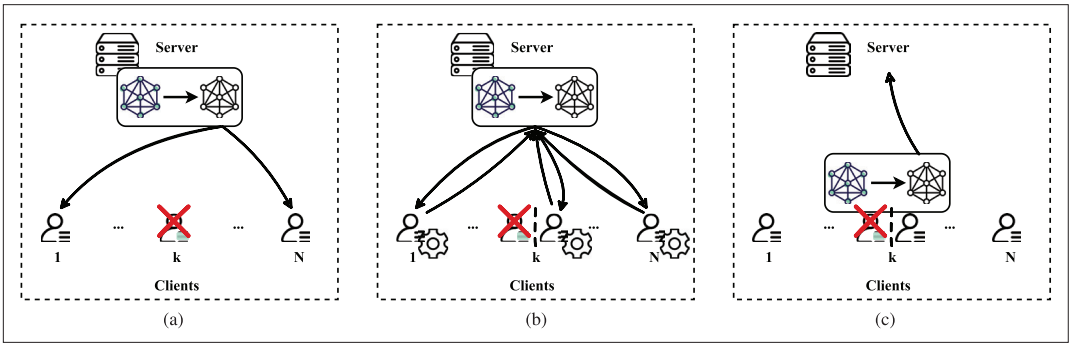

FIGURE 2. Different roles that carry out the unlearning process: a comparison. a) The server. b) All (or the remaining) clients. c) The target client.

While other existing attacks such as model inversion attacks [9] and gradient leakage attacks [10] are also possible in the context of federated unlearning, we argue that it is more important to examine the intrinsic susceptibility of federated unlearning particular to the MI attacks, given the discrepancy between different versions of models before and after the unlearning process.

Membership Inference Attacks

Black-Box vs. White-Box MI Attacks. In black-box inference attacks, the adversary can only obtain the output of the model on an arbitrary input data but has no access to the model parameters. In white-box attacks, in contrast, the adversary also knows the model parameters and can obtain all the intermediate outputs of the model. White-box attacks are feasible in the context of federated unlearning, since the model architecture is generally known to both the server and the participating clients. Nasr et al. [8], for example, took advantage of the stochastic gradient descent (SGD) algorithm used to train the models to launch effective white-box MI attacks.

Passive vs. Active MI Attacks. Most MI attacks are passive, which means the adversary only observes and infers the available model without making any modification to the learning process. In active MI attacks, the adversary participates in the training process and actively modifies the target model to better suit its attack. In the context of federated learning, the adversary can be the central server or one of the clients participating in the federated learning. We will mainly focus on the honest-but-curious server adversary for our discussions in this paper, who stores and processes clients' updates separately without modifying the model or interfering with the learning process. This can be considered a passive MI attack.

Privacy Leakage From Model Updating

If there is an update in the training data, the resulted models before and after the update should reveal some information on the difference. Making use of both the original model and the updated model, compared to having only a single model, one can improve the effectiveness of MI on the updated training dataset. Many ML models inherently leak information during the model updating process and the diverse information of the updating dataset can be inferred [11], [12].

Threat Model. The threat model of the MI attack with model updates is illustrated in Fig. 3. It is considered that new data joins in the training dataset for machine learning model training over time [12]. For example, the training dataset is updated with new data D', which is disjoint with the original data D. In this case, two models are generated from the training, M and M'. The adversary can take the models as black-boxes and observe the output of each model by querying data examples. With the output posteriors, the adversary infers if a data example belongs to the update set $D \cup D'$.

Adversary's Knowledge. There are two key assumptions on an adversary's knowledge [11], [13]: (1) the target model architecture, and (2) a local shadow dataset from the same distribution as the target dataset. An adversary needs them to train a shadow model to mimic the behavior of the target model, generating training data for the attack model. Nevertheless, these papers have also shown by empirical results that, even without access to data of the same distribution and same model architecture, the attack can still maintain its effectiveness to some extent.

Workflow. The general attack pipeline consists of three phases: the adversary (1) generates the posteriors of the two versions of target models M and M' by querying a target data example x; (2) aggregates the two posteriors to construct the feature [13] (or encodes and decodes the posterior difference to generate update set information [11]); and (3) inputs the constructed feature to the attack model to distinguish if the target data example is in the update set $D \cup D'$.

Model Updating vs. Model Unlearning. Chen et al. [13] focused on investigating MI attacks in the context of machine unlearning [13], where the machine learning model erases some specific training examples that are removed from the training dataset. Having access to the original and the unlearned model in machine unlearning is closely related to having access to the original model and the updated model when updating the training dataset, if the unlearning process is performed by retraining from scratch [12]. Taking Fig. 3 as example, the updated training dataset in this case will become the unlearned training dataset $D \backslash D'$ instead, where D' includes all the data the model has to unlearn.

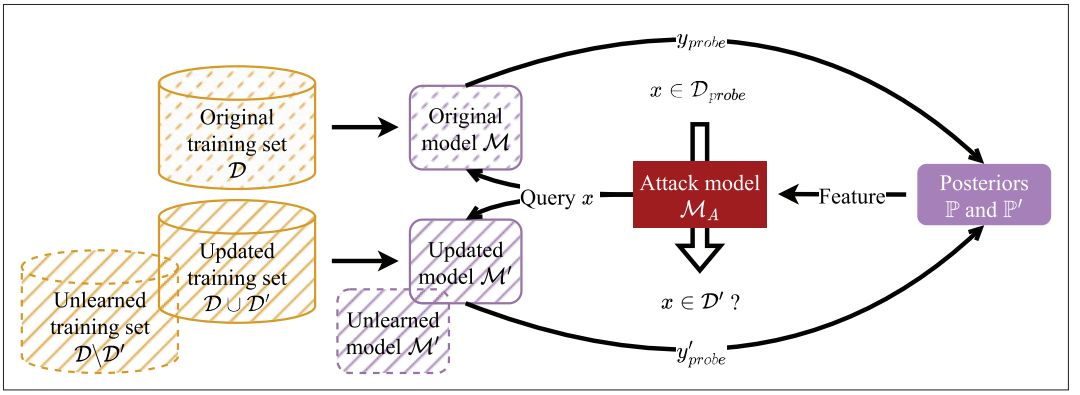

FIGURE 3. Threat model: MI with a single model update (or unlearning, as depicted in dashed containers).

Membership Inference Attacks in Federated Unlearning

We are curious about whether existing MI attacks with model updating are still effective in the context of federated unlearning, where model training is distributed and the training data is kept locally.

As we have mentioned above, the threat model in [11] and [13] has several primal assumptions: the adversary has access to the target model before and after the updating/unlearning as blackbox, the local shadow dataset of the adversary and the target dataset are of the same data distribution, and the shadow model and the target model are of the same structure. Let's think about whether these assumptions are practical in the context of federated unlearning if the honest-but-curious server is the adversary, which naturally knows the model architecture and can build the shadow model accordingly.

Knowledge About the Target Data Distribution. Chen et al. [13] assumed (1) a shadow dataset of the same distribution of the target dataset or (2) a shadow dataset of an arbitrary distribution without hurting performance, in the context of centralized machine learning; and Nasr et al. [8] even assumed (3) the dataset accessible to the attacker partially overlaps the target dataset, in the context of federated learning. Given that it's very common for clients' data distributions to be non-i.i.d. in federated learning, assumption (1) is not practical for any participants including the server or another client as an adversary, while assumptions (2) and (3) may hold for an honest-but-curious server adversary in federated learning as the server can somehow obtain some data samples if it knows the learning task. For example, if clients are collaboratively training a model for image classification, the server is able to train the attack model with a common image dataset from the Internet.

Access to the Original and Target Models. As the honest-but-curious server adversary knows the model architecture, if the target client is the unlearner (as shown in Fig. 2c), the server can easily construct the target client's local model before and after unlearning, and apply an enhanced MI attack by exploiting the intermediate computations of the model. On the other hand, if the server or all the clients are the unlearners (as shown in Fig 2a and b), the global retraining process will introduce lots of stochastic contributions from the other participating clients, in which case the discrepancy between different versions of models is no longer resulted by the deletion of the target client's training data only. This leads to an interesting insight that there may exist a tradeoff between the unlearning spread and the privacy leakage through the process. To be more specific, more participants in the unlearning process lead to extra computation and communication costs, while it becomes harder for the attacker to attribute the discrepancy between the updated models to any individual contribution.

Effectiveness of MI When Federated Unlearning Happens. On top of the discussions on privacy leakage in model updating and its potential adaptation to the context of federated unlearning, there is a concern on the effectiveness of the MI attack when federated unlearning takes place. For machine learning, machine unlearning, or machine learning with model updates mentioned previously, MI attacks are usually conducted when the models have converged and can produce different patterns of posteriors for seen or unseen data samples. However, it's more common in federated learning that the unlearning happens at any time, or even way before the global model converges, under which circumstances the MI attacks might be ineffective on these models that are yet to converge.

Defenses

We are now ready to summarize existing defense mechanisms against traditional MI attacks or enhanced MI attacks with updated models in machine learning, and to investigate whether these defenses are still valid or effective for enhanced MI attacks in federated unlearning.

Defense mechanisms for membership inference attacks can be generally categorized in two ways as shown in Table 2: reducing the adversary's knowledge, or reducing the impact of a single sample on the output of the models. Several defenses have been examined by [7] and [13] against their proposed inference attack.

Confidence score masking is used to hide the true values of the posteriors returned by the target models [7]. There are three masking methods: providing only the top-*k* confidence scores instead of the complete prediction vector to the attacker, providing only the prediction label, and adding noise to the prediction vector. These confidence score masking methods do not need to modify the target models, and thus will not affect the models' accuracy. Publishing only the top-*k* confidence values of the posteriors returned by both original and unlearned models fails to mitigate the attack proposed in [13]. Publishing only the label can effectively mitigate the attack as deleting one data sample in the training dataset is unlikely to

Category	Defense	Against attack [13]	In FL
Reducing the adversary's knowledge	Publishing only the top *k* confidence values	Ineffective	No longer valid
	Publishing only the label	Effective	
	Adding crafted noise to posteriors	Not examined	
Reducing the impact of a single sample on the output of the models	Temperature scaling	Effective but requires softmax as the last layer	Valid
	Differential privacy	Effective but degrades model's accuracy	Valid but affects accuracy dramatically
	Regularization		Valid

TABLE 2. Different defense mechanisms against membership inference attacks and their effectiveness when facing the specific enhanced MI attack [13] or federated learning.

make a change to the output label of a specific target sample. However, these defense methods are no longer valid in the context of federated unlearning, since the adversary has access to the white-box global model and thereby the entire confidence values of the posteriors.

Temperature scaling, which divides the logit vector by a learned scaling parameter, is also an effective defense. One limitation is that it is only applicable to neural networks whose last layer is softmax. Differential privacy (DP), which is commonly used against inference attacks in machine learning, can still effectively prevent the enhanced attack in [13]. However, DP is likely to dramatically affect the model's accuracy after training converges, especially in federated learning.

In addition to confidence score masking and differential privacy, a survey about membership inference attacks on machine learning [14] has summarized several other defenses. One category of methods worth mentioning is regularization. MI attacks are effective especially when the model is fully converged and overfitting. Therefore, methods that can reduce the overfitting level of the target models will hinder the success of the attack. Regularization, such as L2-norm regularization [7] proposed to improve the generalizability of a learned model, can work well in this situation. Compared with confidence score masking methods, regularization methods modify not only the posterior distribution of the target models but also the model parameters, which, however, may inevitably influence the model's accuracy as a result.

Overall, whichever defense against MI attacks is used, there exists a membership privacy-utility tradeoff where the MI attack effectiveness is reduced while the target models' accuracy is also impacted.

Privacy-Preserving Federated Unlearning Framework

With all the findings and analysis presented so far, we attempt to argue what a privacy-preserving federated unlearning framework should be like in this section. Though we do not have a concrete idea or algorithm yet, we show some insights that may benefit future work.

The Target Client Should be the Unlearner. As we have discussed, the unlearner can be the server, the remaining clients or all clients, or the target client, as we categorized from existing federated unlearning algorithms. While they have different strengths and drawbacks due to different capabilities, access to data, etc., we have a preference for the target client as the unlearner if we are concerned about privacy leakage. As more parties participate in the data unlearning at one client, there will be more information sharing intuitively. For example, if the server participates, it needs to store the history of parameter updates, which is harmful to clients' privacy since the curious server can exploit those updates history to even reconstruct data. When performed only at the target client, the unlearning process can be more flexible as the target client can do it anytime at its own will and send the unlearned model to the server afterwards. It does not require other clients to collaborate with extra computation or communication costs. In addition, the target client can verify the unlearning performance locally to make sure the data is forgotten before sending the unlearned model for further aggregation.

Defenses Against MI Attacks Should be Integrated into Federated Unlearning Algorithms. The enhanced membership inference attacks [13] exploit the posteriors of models before and after dataset updating to improve inference accuracy. We've argued that in federated unlearning, there inherently exist such two versions of models, before and after unlearning, available to the aggregating server. When the target client is the unlearning performer, as shown in Fig. 4, the two versions of models are the last global model before local unlearning and the new local model after unlearning to be sent to the server. The adversary at the server could conduct the enhanced MI attack using these two models and infer if some specific data points belong to the dataset that was unlearned, which means the data unlearning cannot really erase the data effectively.

Therefore, we need to integrate effective defenses into federated unlearning algorithms to avoid information leakage by enhanced membership inference attacks. Confidence score masking methods may no longer be valid in this situation since the adversary knows the architecture and the parameters of the two versions of models. Differential privacy with sufficient levels of privacy guarantees in federated learning may incur a significant amount of degradation to the global model's accuracy. The remaining defenses discussed in Section IV, such as temperature scaling and regularization, can be considered to be added to the federated unlearning algorithm as well, but the tradeoff between model robustness and convergence performance still needs to be balanced. Furthermore, it is important to consider additional defense techniques against other potential attacks such as model inversion or gradient leakage attacks that can be integrated into the federated unlearning process.

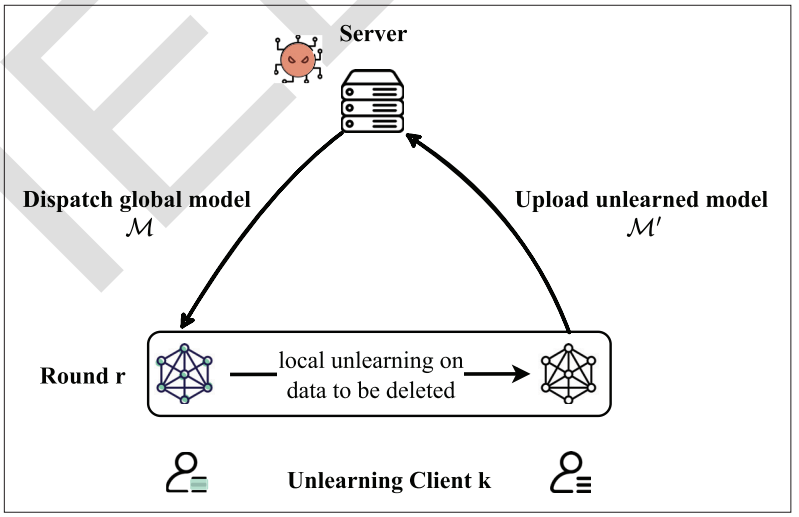

FIGURE 4. The two versions of models available to the adversary at the honest-but-curious server with the target client as unlearning performer: the global model before local unlearning and the local model after local unlearning (the same global model adding the local model updates).

Concluding Remarks

In this article, we provide a comprehensive investigation into recent advances in federated unlearning, including its potential privacy risks involving membership inference attacks, as well as potential defenses against them. We introduce and categorize state-of-the-art mechanisms in federated unlearning and compare their strengths and drawbacks in terms of different unlearning targets and unlearning performers. We are the first to notice the information leakage risk induced by the discrepancy between the models before and after unlearning in the context of federated unlearning, and present our findings on the inherent vulnerability of the federated unlearning paradigm to membership inference attacks. We discuss the potential of widely-used defenses against membership inference attacks and provide suggestions for improving the privacy-preserving nature of federated unlearning mechanisms.

As future work, we believe that the potential of information leakage in federated unlearning will be a critical challenge, as it defeats the purpose of performing the unlearning process in the first place. There is also a need for improving resilience against enhanced membership inference attacks and other privacy attacks. In addition, Gupta et al. [15] proposed a new differentially private machine unlearning mechanism for streaming data removal requests, but applications of differential privacy have still not been considered in the context of federated unlearning. It would be theoretically interesting to study how resilent this category of unlearning mechanisms will be against privacy attacks.

Acknowledgment

The work was supported in part by RGC RIF grant under Contract R6021-20; and in part by RGC GRF grants under Contract 16209120, Contract 16200221, and Contract 16207922.

References

[1] A. Ginart et al., "Making AI forget you: Data deletion in machine learning," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 1–14.
[2] L. Bourtoule et al., "Machine Unlearning," in *Proc. 42nd IEEE Symp. Secur. Privacy (S&P)*, May 2021, pp. 141–159.
[3] G. Liu et al., "FedEraser: Enabling efficient client-level data removal from federated learning models," in *Proc. IEEE/ACM Int. Symp. Quality Service (IWQoS)*, Jun. 2021, pp. 1–10.
[4] Y. Liu et al., "The right to be forgotten in federated learning: An efficient realization with rapid retraining," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2022, pp. 1749–1758.
[5] C. Wu, S. Zhu, and P. Mitra, "Federated unlearning with knowledge distillation," 2022, *arXiv:2201.09441*.
[6] A. Halimi et al., "Federated unlearning: How to efficiently erase a client in FL?" in *Proc. Workshop Updatable Mach. Learn. (UpML)*, 2022, pp. 1–7.
[7] R. Shokri et al., "Membership inference attacks against machine learning models," in *Proc. 38th IEEE Symp. Security and Privacy (S&P)*, May 2017, pp. 3–18.
[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. 40th IEEE Symp. Secur. Privacy (S&P)*, May 2019, pp. 739–753.
[9] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM Conf. Comput. Commun. Secur. (CCS)*, Oct. 2015, pp. 1322–1333.
[10] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. 33rd Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 1–11.
[11] A. Salem et al., "Updates-Leak: Data set inference and reconstruction attacks in online learning," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1291–1308.
[12] M. Jagielski et al., "How to combine membership-inference attacks on multiple updated models," 2022, *arXiv:2205.06369*.
[13] M. Chen et al., "When machine unlearning jeopardizes privacy," in *Proc. 28th ACM Conf. Comput. Commun. Secur. (CCS)*, 2021, pp. 896–911.
[14] H. Hu et al., "Membership inference attacks on machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 1–37, 2022.
[15] V. Gupta et al., "Adaptive machine unlearning," in *Proc. 35th Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 16319–16330.

Biographies

Fei Wang (Student Member, IEEE) (silviafey.wang@utoronto.ca) received the B.Eng. (Hons) degree from the Hongyi Honor College, Wuhan University, China, in 2020. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Toronto, Canada. Her research interests lie at both efficiency improvement and privacy leakage in distributed machine learning. She was a recipient of the Best Paper Award with IEEE INFOCOM 2023.

Baochun Li (Fellow, IEEE) (bli@ece.toronto.edu) received the B.Eng. degree from Tsinghua University in 1995 and the M.S. and Ph.D. degrees from the University of Illinois at Urbana–Champaign in 1997 and 2000, respectively. Since 2000, he has been with the Department of Electrical and Computer Engineering, the University of Toronto, where he is currently a Professor. Since 2005, he has been with the Bell Canada Endowed Chair in computer engineering. His current research interests include cloud computing, security and privacy, distributed machine learning, federated learning, and networking. He was the recipient of IEEE Communications Society Leonard G. Abraham Award in the field of communications systems in 2000, Multimedia Communications Best Paper Award from the IEEE Communications Society in 2009, and the University of Toronto McLean Award. He is a member of ACM. He is a Fellow of the Canadian Academy of Engineering.

Bo Li (Fellow, IEEE) (bli@cse.ust.hk) received the B.Eng. (summa cum laude) degree in computer science from Tsinghua University and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts at Amherst. He is currently a Chair Professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Between 2010 and 2016, he was a Cheung Kong Visiting Chair Professor with Shanghai Jiao Tong University and the Chief Technical Advisor for ChinaCache Corporation, a leading CDN provider, and an Adjunct Researcher with Microsoft Research Asia from 1999 to 2006 and Microsoft Advanced Technology Center from 2007 to 2008. He made pioneering contributions in multimedia communications and the Internet video broadcast, in particular CoolStreaming system, which was credited as first large-scale peer-to-peer live video streaming system in the world. It attracted significant attention from both industry and academia. He has been an Editor or a Guest Editor of more than two dozen of the IEEE and ACM journals and magazines. He was the Co-TPC Chair of the IEEE INFOCOM 2004. He was the recipient of the Test-of-Time Best Paper Award from IEEE INFOCOM 2015 for his contributions.

In the context of federated learning, the adversary can be the central server or one of the clients participating in the federated learning.

These diverging assumptions intrinsically determine the limitations of their algorithms.

This defeats the purpose of federated unlearning, where such private data are supposed to be deleted and protected from further exposure.