A Taxonomy of Attacks on Federated Learning

Malhar Jere, Tyler Farnan, and Farinaz Koushanfar | University of California, San Diego

Federated learning is a privacy-by-design framework that enables training deep neural networks from decentralized sources of data, but it is fraught with innumerable attack surfaces. We provide a taxonomy of recent attacks on federated learning systems and detail the need for more robust threat modeling in federated learning environments.

The maturation of the information age has enabled explosive growth in the field of machine learning. Deep learning (DL) is an advanced type of machine learning, responsible for many of the modern breakthroughs that are popularizing the field. DL models use hierarchical feature extraction to represent complex underlying patterns in observed information, and they can learn to make highly accurate predictions across a variety of unrestricted input spaces, such as images, audio, and natural language. For some narrow tasks in image recognition and game playing, DL models can outperform human benchmarks. Most often, the success of a DL project relies on two core components: 1) a researcher's ability to collect a very large set of training data and 2) ensuring that the learning objective is well represented by the training data.

As the number of intelligent devices and applications grows, the volume and diversity of data generated at the network "edge" will continue to increase at astounding rates. This abundant source of data is inherently decentralized and heterogenous, and when analyzed in aggregate, it may contain insights and discoveries that have the potential to accelerate scientific and technological innovation. However, the privacy risks associated with data ownership are quickly becoming a focal point of public awareness. The tension between high-quality services and user privacy is driving demand for novel research and technology to enable methods for extracting insights from data without exposing the privacy of that information.

One novel approach that has the potential to alleviate this tension is federated learning. It is a type of distributed machine learning that enables a network of client devices to collectively train a DL model. A key characteristic is that the training data remain stored on the devices that originally recorded or generated them. By conducting model training at the network edge, aggregate analytics can be achieved without the need to amass the data themselves. In practice, there is a critical and challenging tradeoff between the privacy of training data and the utility of the model. Realizing the appropriate tradeoff is highly dependent on the use case, including factors such as the type of data, model architecture, and intended application. Beyond fundamental and domain-specific constraints, the security and privacy of the federated learning infrastructure itself are critical to designing collaborative learning networks that clients can trust.

To date, the security of traditional DL systems has been largely studied in the context of adversarial attacks

Digital Object Identifier 10.1109/MSEC.2020.3039941
Date of current version: 28 December 2020

during the inference time, for example, fooling a predictive service to avoid an unwanted yet correct classification. However, federated learning systems expose several new attack vectors at training time, giving rise to complex, open challenges surrounding the following core questions:

- How can one guarantee the robustness of a collaboratively trained DL model?
- How can one preserve the privacy of client data contributions?
- How can one protect the intellectual property embodied by a DL model?
- From a central server's point of view, how can one trust the reported results of participants in a network?
- From an individual participant's point of view, how can one trust the central server?
- How can one ensure that the distributed neural network architecture is robust to backdoor attacks?

While some of these questions have been discussed in prior literature,[1] our work is different in that we seek to address highly specific security and privacy concerns related to federated learning and propose several solutions to counter these threats. Going forward, addressing each of these open questions around the security, privacy, and effectiveness of federated learning systems will have significant societal impacts.

The purpose of this article is to provide a comprehensive overview of attacks on federated learning systems, with respect to both model performance and data privacy. We present an overview of federated learning (see "Glossary of Parameters Required for Robust Federated Learning Threat Models") and establish a framework for robust threat modeling. We then survey model performance and data privacy attack types, and we assess some of the existing defense methods. Finally, we present a taxonomy of adversarial federated learning and highlight the open challenges and opportunities in this important and emergent field of collaborative machine learning.

Glossary of Parameters Required for Robust Federated Learning Threat Models

Most, if not all, existing attacks on federated learning systems can be categorized as compromising either model performance or data privacy. Each attack can be characterized in terms of a number of parameters of the federated learning system, such as the number of compromised users, number of total users, aggregation scheme used by the central server, and differential privacy parameters of each user. We outline each of the individual parameters in the following:

- The number of participants (N_i) is the number of federates participating in round i.
- The total amount of data (D) is the aggregate of the training data on all federated devices.
- The fraction of data (f_i^j) is the fraction of the total data D held by user j in round i.
- The diversity of data (d_i^j) is the distribution of data in f_i^j.
- The model architecture (M) held by each federate is, in most scenarios, constant across devices.
- The participation rate (p_i^j) concerns each participant j in the federated averaging algorithm at round i, which may randomly choose users for security.
- The adversarial risk (r_i^j) concerns each participant j in the federated averaging algorithm at round i, which may randomly choose users for security. Adversarial risk here denotes the probability that a particular user may engage in an attack that may compromise data privacy or the integrity of the global model. The adversarial risk of individual users can be calculated based on the behaviors of users during the training process.
- The client behavior (u_i^j) concerns each participant j in the federated averaging algorithm at round i. The behavior of users may be benign or malicious, and it may not be constant through time (benign users could suddenly become malicious, and vice versa).
- The access parameter (b_i^j) concerns each participant j in the federated averaging algorithm at round i, which determines whether adversaries have white-box or black-box access to their federate model. White-box attacks are generally easier to launch.
- The adaptability (a_i^j) of each participant j in the federated averaging algorithm at round i denotes whether a participant is static in his or her attack or launching multiple attacks at once.
- The behavior (s_i^j) of the server s in the federated averaging algorithm at round i may be benign or malicious, and it may not be constant in time (benign servers could suddenly become malicious, and vice versa).

Federated Learning

Background

Google keyboard's next-word prediction was recognized as the first production-grade federated learning system for mobile devices.[1] The principal goal of this system was to minimize the network communication overhead. This article introduces the canonical federated averaging algorithm, which is the core method to compute the global model update from an aggregation of all the local updates provided by participating clients. This method greatly minimizes the volume of data that are exchanged since only model parameters need to be traded to learn from a distributed and private data set, rather than exchanging entire data sets.

Network Types

Federated learning can occur in a range of network types,[1] privacy settings, and objectives, and it can be divided into two broad categories: cross-device versus cross-silo. In cross-device learning, many client devices (sometimes millions) are users of a central provider's intelligent service. The client data sets are usually small, and connectivity to the network is highly intermittent. In cross-silo learning, a small number of clients (tens to hundreds) cooperate to achieve shared objectives. In this case, local data sets are much larger, connectivity is more reliable, and the client's compute resources are likely more powerful. Both the security and performance threats and advantages of a federated learning system will largely be defined by the network type in addition to other more complex criteria specific to the machine learning objective, privacy constraints, and chosen methods.

New Attack Surfaces

In traditional machine learning projects, the development process (the learning phase) is protected and centralized within a single organization. As a result, the secure-machine-learning literature has largely focused on defending against adversarial clients during inference on a developed model in production. Federated learning is a new type of distributed learning system that is vulnerable to many underexplored attacks on data privacy and model security during the learning phase.

The largest threat surface on a federated learning system is the network of clients that participates in the learning phase by contributing data. Such client-side attacks are perhaps even more powerful than inference phase effects because, rather than simply exploiting the boundaries of a model service in production, adversarial federated learning clients are in a position to manipulate and shift the boundaries of the model during development. We provide a list of existing attacks on federated learning systems in Table 1.

Table 1. The threat modeling of existing literature attacks.

Attack	*N*	*A*	*D*	Compromised behavior?	Data sets
Local model poisoning attacks	100	20	0.5	Test accuracy	MNIST, fashion-MNIST, Wisconsin Breast Cancer data set
Distributed backdoor attacks	10	1	0	Model back door	Loan data set, fashion-MNIST, CIFAR-10, Tiny ImageNet
Exploiting unintended feature leakage	100	1	0	Model back door	CIFAR-10, public Reddit data set
Fall of empires	25	11, 12	1	Malicious gradients	CIFAR-10
How to backdoor federated learning	10	1	0	Model back door	CIFAR-10, public Reddit data set
Analyzing federated learning through an adversarial lens	10	1	1	Model back door	Fashion-MNIST, UCI adult census
Can you really backdoor federated learning?	30	11	0	Model back door	EMNIST
Contamination attacks and mitigation in multiparty machine learning	10	1	1	Data poisoning	UCI adult, UCI credit card, News20
A little is enough.	51	12	1	Model and data poisoning	CIFAR-10, fashion-MNIST, CIFAR-100
Mitigating sybils in federated learning poisoning	10	2	1	Model and data poisoning	MNIST

Parameters N, A, *and* D, *which are derived from the glossary, denote the number of active users, number of adversarial users, and data set diversity, respectively. MNIST: Modified National Institute of Standards and Technology; CIFAR: Canadian Institute for Advanced Research; UCI: University of California, Irvine; EMNIST: Extended MNIST.*

Secure Aggregation

To date, a central point of focus in the federated learning community has been on designing methods for secure aggregation, aiming to address the threat of an untrusted central server. The purpose of a secure aggregation protocol is to prevent the central server from gaining visibility and control over a client's local models while executing the federated averaging algorithm. The backbone of secure aggregation is secure multiparty computation, a branch of cryptography that enables untrusted parties to cooperatively evaluate a function on hidden inputs. This is a highly active area of research, and numerous techniques have been proposed for federated learning.[1] A related approach leverages trusted execution environments, which are hardware-based solutions serving as "reverse black boxes"; i.e., no one can observe the inputs and computation inside a secure enclave.[2]

Survey of Model Performance Attacks

Model corruption is a "training time" threat that can be separated into two types of adversarial objectives: untargeted and targeted poisoning attacks. An untargeted poisoning attack[3] aims to induce a denial of service by degrading the overall performance of a given model. Targeted attacks, also known as *backdoor* and *trojan* attacks,[1] aim to manipulate the model to learn a malicious misclassification of samples from a particular distribution while also maintaining high accuracy on the global objective. Figure 1 highlights how an adversary on an individual device might corrupt the global federated learning system by manipulating updates. Figure 2 illustrates how attacks and defenses might be potentially taxonomized based on an experimental setting.

Both targeted and untargeted attacks can be realized by two poisoning methods: data poisoning and model poisoning. Data poisoning attacks aim to maliciously manipulate training and/or inference samples, indirectly inducing model corruption. Model poisoning attempts to directly induce model corruption by manipulating the learning process itself.

Data Poisoning

Data poisoning attacks on federated learning systems typically compromise the integrity of the training data so as to compromise the performance of the model, such as inserting a back door for certain triggers during inference and by reducing the accuracy of the overall model. Several different methods of data poisoning have emerged in recent literature, as summarized in the following:

- *Label flipping*: Adversaries with access to training data permute labels while keeping the features intact.[1] Prior

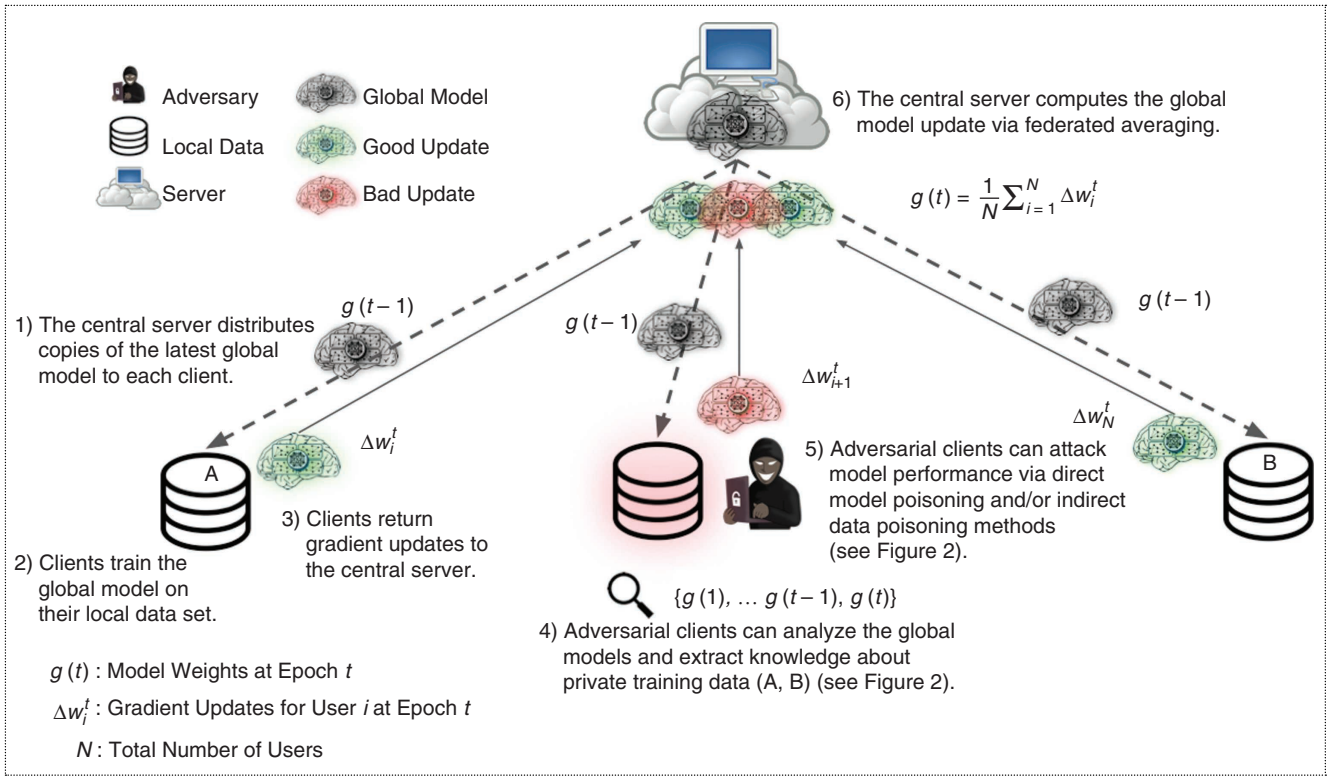

Figure 1. An overview of a single round of federated learning with an adversarial client.

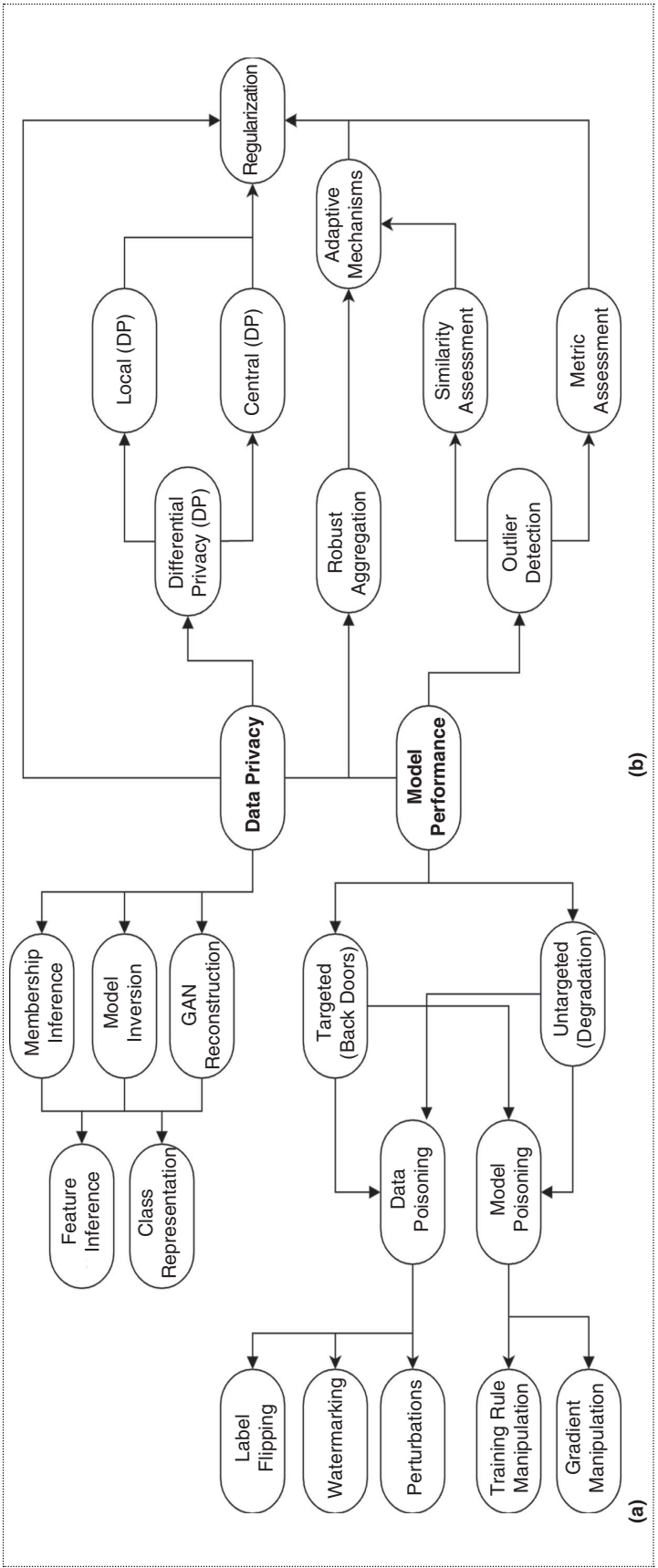

Figure 2. A high-level taxonomy of (a) the primary attack methods and (b) the defense methods for both client-driven model performance and data privacy threat types.

attacks on the Modified National Institute of Standards and Technology data set have involved flipping labels for sevens into ones, and vice versa, while preserving the image content itself, thereby intentionally fooling federated learning models.

- *Backdoor insertion*: Adversaries can manipulate training data such by adding stamps or watermarks, thereby enabling models to learn triggers on inputs with those manipulations while still preserving accuracy on clean data, thus making attacks harder to detect.[1]

Model Poisoning

Model poisoning involves a broad category of methods to manipulate the training procedure for federated learning. While data poisoning attacks may be considered a subset of model poisoning, other attacks, such as directly manipulating gradients and changing the learning rule, fall under the umbrella of model poisoning, as described in the following:

- *Gradient manipulation*: Adversaries can manipulate local model gradients to compromise the global model performance, for instance, by reducing the overall accuracy.[1] Bagdasaryan et al.[21] inserted hidden global model back doors by direct gradient manipulation.
- *Training rule manipulation*: Several other works involve manipulating model training rules[1] and demonstrate that model poisoning through this method is far more effective than data poisoning; in some cases, even manipulating a single local model can compromise the global model. One example of advanced training rule manipulation is reported by Bhagoji et al.[22] The authors successfully achieve a stealthy targeted model poisoning attack by adding a penalty term to the objective function to minimize the distance between malicious and benign weight update distributions.

Survey of Data Privacy Attacks

Federated learning has attracted attention from a number of communities seeking to conduct large scale analysis on sensitive data sets. While federated learning enables cooperative machine learning without transferring data sets to an external service, it does not prevent information leakage from the DL model parameters. Machine learning models are generally vulnerable to attacks intended to extract information about the training data via interaction and the analysis of a trained model's parameters.[1] As a result, various privacy attacks have been demonstrated to extract meaningful insights from training data stored in the model parameters themselves. These types of attacks can be classified as model inversion,[4] membership inference,[5] and generative adversarial network (GAN) reconstruction.[6]

Furthermore, various studies of these attacks have demonstrated significant vulnerabilities in differential

privacy, which is a well-adopted technique that will be discussed in a later section about defense methods. Recall that in federated learning, since all clients receive a copy of the global model during each round, any inference phase attack on data privacy is viable during the learning phase, as well. However, preventing learning phase data extractions is significantly more challenging. An effective learning phase defense mechanism needs to account for continuous, distributed interactions with the model during its development.

Model Inversion Attacks

Model inversion attacks have been shown to successfully characterize sensitive properties of the classes and/or individual samples represented by the model.[1] Given white-box access to a model, Fredrikson et al.[23] report that model inversion attacks on decision trees can identify instances of sensitive variables (survey responses) with no false positives. As another example, in a study on pharmacogenetics, an attacker can predict a patient's genetic markers, given just the model and some basic demographic information about the targeted individual.[1]

Membership Inference Attacks

Membership inference attacks seek to determine whether a given sample was included in the training data set and/or whether a given sample belongs to a specific class represented by the model. Furthermore, this type of attack can succeed even when the objective is uncorrelated with primary features of the class.[8]

GAN Reconstruction Attacks

GAN reconstruction attacks resemble model inversion attacks; however, they are demonstrably more powerful and have been shown to generate synthetic samples that are statistically representative of the training data.[6] Hitaj et al. show that when attacking more complex DL architectures, such as convolutional neural networks, traditional model inversion methods completely fail, while GAN reconstruction attacks can successfully generate realistic samples, even when moderate differential privacy is applied to the training procedure. Additionally, while model inversion requires only accessing the discriminator model after training, GANs jointly train the generator and the discriminator. Hence, during collaborative learning, it is possible for an adversary to influence benign clients to unknowingly commit gradient updates that leak more private information than intended.[6]

Survey of Defense Methods

Recent research into federated learning security has aimed to stress-test existing methods for preventing model corruption and private information extraction. Fundamentally, these defense objectives share some underlying properties and can be related to fundamental machine learning theory about generalization. In essence, when a model generalizes well, it has learned to represent the underlying statistical distribution from which its training examples were generated. When a model has not generalized, one common outcome is referred to as *overfitting*. Overfitting can be thought of as "memorizing" the training data, and it is typically indicated by a significantly higher accuracy with training data than with unseen (test) data.

When preventing model corruption, a defense method seeks to prevent the model from overfitting to malicious updates, minimize the overall influence of potentially malicious updates, and prevent malicious updates from being incorporated into the global model. Some methods defending against private information leakage, such as adversarial regularization techniques,[9] attempt to prevent the model from overfitting or "memorizing" distinguishing information about any given data point. Indeed, there is an interesting yet underexplored relationship between generalization and privacy preservation.[1] We will briefly highlight the prominent types of defense methods: differential privacy is the standard for privacy preservation, while aggregation mechanisms and outlier detection are primarily oriented toward mitigating the impact of model corruption attacks.

Differential Privacy

Differential privacy is a formalized mathematical guarantee that quantifies the amount of information revealed about a single entry in a database. For example, a privacy-constrained statistician would use differential privacy to bound the measurable difference between two adjacent data sets (with exactly one entry missing). The objective of differential privacy, then, is to provide a guarantee that, with high probability, no single record in a data set can be meaningfully distinguished from the rest of the records.[11]

In federated learning systems, differential privacy can be used to minimize the risk of information leakage from the DL model parameters. Most commonly, statistical noise can be added to the output of the algorithm used to query or analyze a client's data set. Additionally, the information contributed by clients can be bounded by regularization techniques, which attempt to exclude outliers such that the aggregated update bounds the variance in the global distribution. In practice, differential privacy can be implemented in a variety of ways. For example, in local differential privacy, each client adds its own privacy prior to broadcasting its update in the network.[1] In contrast, central differential privacy relies on a trusted server to implement privacy transformations on behalf of all clients. When considering model performance attacks, Bagdasaryan et al.[21] demonstrate that local differential privacy can successfully defend against backdoor attacks, but the required noise levels significantly hinder

the model's learning ability. Further research is required to explore the extent of differential privacy's utility as a defense method beyond privacy preservation alone.

Robust Aggregation

Aggregation algorithms are central to federated learning. There is a breadth of research on Byzantine-robust algorithms that seek to sustain high-performance model development when faced with network instabilities, such as client dropout, erroneous updates, and malicious actors. Many of the foundational Byzantine-robust aggregations, such as Krum, Bulyan, the trimmed mean, and the coordinate-wise median,[1] rely on theoretical assumptions that do not hold for federated learning, and their vulnerabilities have been empirically evaluated in prior work. For example, Bagdasaryan et al.[21] demonstrate that attackers controlling 1–10% of the clients can be selected by the Krum aggregation algorithm 80–97% of the time, respectively.[1]

As a result, a number of adaptive aggregation mechanisms has been proposed, such as adaptive model averaging for rejecting systematically corrupt client updates[1] and residual-based reweighting to reduce the effect of data poisoning attacks.[12] For example, Munoz et al. report that their adaptive federated averaging algorithm can successfully detect Byzantine adversarial clients 90–100% of the time but that it fails to find noisy adversarial clients, with only 2.7–20% success rates.

Outlier Detection

This is a more proactive form of defense that aims to explicitly identify and deny malicious influence on the system. One popular approach, known as *reject on negative impact*, can be generalized for federated learning by measuring the test error of a given update and rejecting it if it doesn't improve the global model.[13] While this may work against untargeted poisoning attacks, it fails to address stealthy and/or targeted backdoor attacks that do not degrade the performance on the global objective. Another detection-based defense, known as *TRIM*, seeks to find a subset of updates that minimizes the loss on the global objective by removing outliers with high residuals.[1] Jagielski et al. report that the TRIM method maintains a mean square error within 1% of that of unpoisoned linear regression models when defending against many types of poisoning attacks on health-care, loan, and housing data.

One interesting type of detection mechanism computes distance measurements between client updates and is a natural fit for federated learning systems, where understanding the heterogeneity of client distributions is critical. One such approach, referred to as *FoolsGold*, adapts client learning rates, based on contribution similarity.[14] A key finding is that in the nonindependent identically distributed setting (which is typical for federated learning), a group of sybils colluding on a malicious poisoning attack will have noticeable statistical similarity compared to the diverse population of benign clients. In a complementary approach, Bhagoji et al.[22] show that when the majority of benign clients are cooperating on a global objective, a malicious update can be distinguished as an outlier by measuring the distance from the global distribution of parameter updates.

Table 2. A list of unsolved security and privacy problems in federated learning.

Attack method	Specific attack type	Defended against?
Data poisoning	Label flipping	✓
	Back door	Partially
	Gaussian	✓
Model poisoning	Training rule manipulation	Seven
	Gradient manipulation	Seven
Model inversion attacks	Class representation	✓
Membership inference	Confidence score based	✓
	Label based	Partially

While this is not an exhaustive list, it highlights several state-of-the-art attacks and defenses in federated learning at the time this article was written.

Urgent Challenges

Federated learning offers a new paradigm to preserve user privacy while enabling effective machine learning on a global scale, but it is fraught with several urgent challenges that must be addressed. Several urgent challenges are highlighted in Table 2 and include the following:

- *Robustness to adversarial attacks*: Neural networks have been shown to be vulnerable to a wide variety of adversarial attacks, such as imperceptible adversarial samples.[1] Adding neural networks in a federated environment leads to further risk as they become ever more widespread. While the question of adversarial robustness is an active research topic, several suggestions to counter this involve 1) devising newer robustness metrics for images,[15] text, and audio; 2) incorporating robustness audits into the deployment process; and 3) lifelong red-teaming deployed models against unseen adversaries.[16]
- *Robustness to privacy attacks*: Neural networks are vulnerable to membership inference attacks[5] through which adversaries can glean sensitive private information from the training data for a given model. An immediate suggestion for prevention would involve gating access to the model and training models via differential privacy and strong L_2 regularization,[17]

which are the only known defense strategies that successfully prevents all such attacks.

Open Questions in Federated Learning

The prospect of unlocking insights from private data while maintaining user privacy offers immense potential in a variety of fields, such as medicine, insurance, health care, education, and more. However, it also leaves several open questions that must be addressed, including the following:

- *Ensuring and building trust*: How can federated learning servers trust reports, such as the test accuracy, the data set size, and others, from individual users? In particular, when data are privately held on devices, there remains little scope for manual verification. Cryptographic protocols, such as zero-knowledge proofs and garbled circuits, may offer promise for secure calculations using private data, but they remain unfeasible on a large scale. Semisupervised methods, such as MixMatch,[18] offer a method to pretrain computer vision models without needing explicit labels. Central servers can then aggregate individual models and fine-tune them on a held-out independent data set.
- *Ensuring robustness against model back doors*: Numerous backdoor attacks have been proposed in machine learning and computer security literature for neural network models. Is there any way to guarantee robustness against back doors through smart model design? Or is the question of avoiding back doors itself circular, as the model simply learns the distribution present in the data set? Checkpointing models, such as through DeepFense,[19] and developing new metrics for robustness against adversarial attacks[20] show promise.
- *Ensuring protection of the intellectual property embodied by a DL model*: Privacy attacks, such as model inversion, membership inference, and GAN reconstruction, have been shown to be effective on dynamic neural networks. How can federated systems ensure that the data embodied by the model are privately held and free from reconstruction attacks?

In conclusion, federated learning presents an exciting opportunity to facilitate collaboration and accelerate innovation without requiring the exchange and centralization of private data. However, significant research and engineering efforts will be required to build trustworthy federated learning systems that are robust against adversarial machine learning techniques. ■

Acknowledgment

This work was supported by Semiconductor Research Corporation.

References

1. P. Kairouz et al., "Advances and open problems in federated learning," 2019, arXiv:1912.04977.
2. M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, 2015, vol. 1, pp. 57–64. doi: 10.1109/Trustcom.2015.357.
3. C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," 2017, arXiv:1703.01340.
4. 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. Security*, 2015, pp. 1322–1333.
5. R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. Secur. Privacy*, 2017, pp. 1–15.
6. B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. Comput. Commun. Secur.*, 2017. pp. 1–13. doi: 10.1145/3133956.3134012.
7. M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. 23rd {USENIX} Security Symp. ({USENIX} Security 14)*, 2014, pp. 17–32.
8. L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 691–706. doi: 10.1109/SP.2019.00029.
9. M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 634–646.
10. S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *Proc. IEEE 31st Comput. Security Found. Symp. (CSF)*, 2018, pp. 268–282. doi: 10.1109/CSF.2018.00027.
11. C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theoretical Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014. doi: 10.1561/0400000042.
12. S. Fu, C. Xie, B. Li, and Q. Chen, "Attack-resistant federated learning with residual-based reweighting," 2019. [Online]. Available: https://arxiv.org/abs/1912.11464
13. M. Barreno, B. Nelson, A. D. Joseph, and J. Doug Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010. doi: 10.1007/s10994-010-5188-5.
14. C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2018, arXiv:1808.04866.
15. M. Jere, S. Herbig, C. Lind, and F. Koushanfar, "Principal component properties of adversarial samples," 2019, arXiv:1912.03406.
16. M. Jere, B. Hitaj, G. Ciocarlie, and F. Koushanfar, "Scratch that! An evolution-based adversarial attack against neural networks," 2019, arXiv:1912.02316.

17. C. A. Choquette Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," 2020, arXiv:2007.14321.
18. D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5049–5059.
19. B. D. Rouhani, M. Samragh, M. Javaheripi, T. Javidi, and F. Koushanfar, "Deepfense: Online accelerated defense against adversarial deep learning," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD)*, 2018, pp. 1–8.
20. M. Jere, S. Herbig, C. Lind, and F. Koushanfar, "Principal component properties of adversarial samples," 2019, arXiv:1912.03406
21. E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.
22. A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Machine Learn.*, 2019, pp. 634–643.
23. 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. Security*, 2015, pp. 1322–1333.

Malhar Jere is a Ph.D. student in the Department of Electrical and Computer Engineering at the University of California, San Diego, California, 92093, USA. He earned his master's in electrical and computer engineering from the University of California, San Diego, in 2019. His research interests include privacy-preserving machine learning and machine learning robustness and security. He is a Student Member of IEEE. Contact him at mjjere@ucsd.edu.

Tyler Farnan is a software engineer at Consensys Health, Wilmington, Delaware, 19801, USA. He received his master's of science in computational science from the Department of Mathematics, University of California, San Diego. His research interests include applications of federated learning to the medical domain. Contact him at tfarnan@ucsd.edu.

Farinaz Koushanfar is a professor and Henry Booker Faculty Scholar in the Department of Electrical and Computer Engineering, University of California, San Diego, California, 92093, USA. She received her Ph.D. from the University of California, Berkeley, in 2005. Contact her at farinaz@ucsd.edu.

Digital Object Identifier 10.1109/MSEC.2021.3060557