A Comprehensive Research of Data Privacy Based on Federated Learning

Junxiang Zhang
School of Advanced Technology, Xi’an Jiaotong-Liverpool University, Suzhou, Jiangsu, 215000, China

Keywords: Federated Learning, Privacy Protection, Attack Methods, Privacy Preservation

Abstract: In recent years, Federated Learning (FL) has gained significant attention as a crucial technology for addressing the issue of data silos. Despite possessing certain privacy-preserving capabilities, FL still carries the risk of privacy leakage, particularly in fields such as healthcare and finance, where the demand for user privacy protection is increasingly urgent. This review first introduces the fundamental principles and classifications of FL, with a focus on discussing its advantages in data privacy protection. Subsequently, it reviews the background of current data privacy challenges, encompassing various privacy attack methods that highlight the deficiencies of FL in privacy protection. Following this, various privacy protection methods are thoroughly discussed, analyzing the strengths of different methods in safeguarding data privacy. A comparative analysis of specific privacy protection algorithms is then conducted, providing a detailed examination of the advantages, disadvantages, protection strategies, and targeted subjects of each algorithm. By systematically summarizing existing research, this paper offers a comprehensive understanding of the application of FL in the field of data privacy, providing valuable insights for both the academic and industrial sectors. Furthermore, it serves as a useful guide for future research and applications in this domain.

1 INTRODUCTION

Federated Learning (FL) has become highly prominent for breaking down data silos, finding applications across finance, healthcare, and smart cities, thereby amplifying the importance of its privacy considerations.

Initially proposed by Mcmahan et al. in 2016, FL is a technology designed for efficiently training high-quality centralized models (Konečný et al. 2016). This technique allows models to be trained on multiple local devices and then centrally aggregated at a central location. Importantly, data is stored on users' local devices rather than being uploaded to a centralized data center, ensuring the privacy of users. Google has made notable contributions to FL, being the first to introduce the concept and providing open-source · frameworks like TensorFlow Federated (TFF) (TensorFlow, 2024).

International standardization organizations, such as the International Organization for Standardization (ISO), and other standardization bodies are actively working on standardizing FL to facilitate its cross-industry applications. For instance, the Institute of Electrical and Electronics Engineers (IEEE) has approved the first standard for FL architecture (IEEE Computer Society 2021). Numerous researchers have focused on studying privacy protection, attacks, and security threats related to FL, proposing various methods to ensure the security of models and data. Examples include the Federated Meta-Learning Algorithm (FedMA), Federated Dynamics Algorithm (FedDyn), Multi-party Optimization with Outcomes Network Algorithm (MOON), and knowledge transfer personalized federated learning (KT-Pfl) algorithm, among others (Wang et al. 2020, Acar et al. 2021, Li et al. 2021, Zhang et al. 2021).

In China, extensive research has been conducted on FL, and the technology has been applied in practical settings, particularly in areas such as agriculture and healthcare, emphasizing privacy protection and model training (Kang et al. 2022, Xu et al. 2021).

FL has successfully addressed the traditional machine learning challenge where uploading all data to a high-performance server for centralized training could lead to issues such as data privacy breaches and uncontrollable data flow. Essentially, FL represents a form of distributed machine learning.

Zhang, J.
A Comprehensive Research of Data Privacy Based on Federated Learning.
DOI: 10.5220/0012832500004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 531-536
ISBN: 978-989-758-690-3

Despite having privacy protection mechanisms, FL remains vulnerable to various attack vectors that may result in the leakage of user data privacy. From the perspective of attack methods, these primarily include poisoning attacks and Byzantine attacks. Regarding the stages of attack initiation, they are broadly categorized into the model training phase and the model inference phase.

2 FEDERATED LEARNING

FL involves collaborative model training by clients under central coordination, with the central server (CS) aggregating locally trained models through weighted averaging to derive a global model (GM) in each iteration. After multiple rounds of iteration, the final result model is achieved. This approach effectively mitigates privacy risks associated with traditional machine learning. Since raw data is stored locally on client devices, only the analysis and sharing of models take place, preventing data leakage to the server or other locations. Additionally, the accuracy achieved is comparable to that of traditional machine learning.

The process involves FL algorithmic principles, focusing on model training in a distributed environment without necessitating raw data transfer to a CS. The basic principles of typical FL algorithms are outlined below:

1. Initialization: Select the architecture and initialize parameters for the GM.

2. Device Registration: Devices register themselves with the FL system.

3. Local Model Training: Each device utilizes local data for model training. Training can involve traditional gradient descent or other optimization algorithms.

4. Model Parameter Update: After local data training, devices transmit only the updates (gradients or weights) of model parameters to the CS, without transferring raw data.

5. Model Aggregation: The CS collects model parameter updates from all devices. Using an aggregation strategy, typically weighted averaging, the new parameters for the GM are obtained.

6. GM Update: The CS updates the GM using the aggregated parameters.

7. Communication and Iteration: Iterate through the process of local model training, parameter updates, model aggregation, and GM updates until convergence or a predefined number of training rounds are reached.

8. Model Evaluation: Evaluate the GM to assess its performance in FL.

It is evident that in FL since clients are responsible for training, they only upload the model without transferring local data. Additionally, the trained model uploaded to the CS can be shared among multiple parties without significantly affecting model accuracy.

3 CLASSIFICATION OF FEDERATED LEARNING

According to different data situations, FL can be divided into three types: Horizontal FL, Vertical FL, and Federated Transfer Learning (Yang et al. 2019). Details are presented in Table 1.

Table 1. Three Types of FL Classification

	User Overlap	**Feature Overlap in Data**
Horizontal FL	Multiple	Few
Vertical FL	Few	Multiple
Federated Transfer Learning	Few	Few

Based on practical production, two scenarios for FL can be defined: Business-to-Business (ToB) and Consumer-to-Consumer (ToC).

In the ToB scenario, the primary entities involved are institutions, companies, and governments. Typically, a third-party CS is used for model exchange and parameter control (Wang et al. 2021).

In the ToC scenario, there is often a larger number of participants with lower computational power. This scenario tends to weaken the characteristics of a CS control node, placing model updates in the hands of each participant (Wang et al. 2021).

4 FEDERATED LEARNING PRIVACY ISSUES

While FL incorporates certain privacy protection mechanisms, it may not provide sufficient privacy safeguards. For instance, attacks during the process of model update data transmission can lead to the leakage of sensitive information. Different attack methods may also result in data leakage from the CS.

4.1 Byzantine Attacks

Byzantine attacks refer to situations in distributed systems where a subset of nodes (called Byzantine nodes) intentionally provides erroneous, deceptive, or malicious information, attempting to disrupt the normal operation of the system. In FL, attackers control multiple Byzantine users who intentionally provide false or harmful parameter data to the CS, disrupting the training process of the GM. This type of attack can impact the GM and compromise its accuracy (Bagdasaryan et al. 2020).

4.2 Poisoning Attacks

A "Poisoning Attack" is where attackers deliberately inject malicious, disruptive, or false data into the FL system to influence the performance of the GM (Chen et al. 2020). Poisoning attacks have various methods, such as data poisoning and model poisoning.

Data poisoning involves contaminating training sample data, such as adding erroneous data or altering local data labels, misleading the GM during training, and disrupting the model's learning of features (Jiang et al. 2019).

Model poisoning disrupts the performance of the GM by injecting malicious local model parameters into the FL system (Bhagoji et al. 2019).

4.3 Sybil Attacks

Sybil attacks typically involve a single node in the network having multiple identity labels and weakening the effectiveness of network redundancy backups through control over the system. Attack methods include direct communication, forgery or theft of identity, and simultaneous and non-simultaneous attacks. In the server-client architecture of FL, participants launching malicious attacks can control the server, forge numerous client devices, or control devices in a pool that have been compromised, enabling the execution of Sybil attacks (Wang et al. 2021).

5 PRIVACY PROTECTION IN FEDERATED LEARNING

Privacy protection of data is a crucial aspect of FL. Without adequate protection, there is a risk of leakage of many privacy parameters during training. Once leaked, both data owners and participants face significant losses. Therefore, it is essential to implement privacy protection measures in FL.

5.1 Defense Against Data Poisoning

Several methods exist to protect learning models from the impact of data poisoning attacks. Examples include anomaly detection, data filtering, and trust evaluation. Nathalie et al. use context information checking to detect toxic sample points. By comparing results from different parts of training, they evaluate and identify abnormal data models (Baracaldo et al. 2017).

5.2 Homomorphic Encryption

Homomorphic encryption is a specialized technique for computational operations on encrypted data. It enables operations such as addition or multiplication on encrypted data without the need to decrypt it, ensuring that the original data remains confidential during transmission. Homomorphic encryption can be utilized to protect model parameters when they are sent from the server to the client in a FL system. This allows clients to update in an encrypted state without exposing model details. During model predictions, homomorphic encryption can be used to encrypt input data, enabling the server to make predictions in an encrypted state without knowing the plaintext content of the input data (Baracaldo et al. 2017).

5.3 Differential Privacy

Differential privacy provides mathematically rigorous protection when handling individual data, preventing re-identification attacks against individual data. It can protect local data on each device by introducing noise, ensuring that even locally, contributions of individual data are not directly exposed, thereby enhancing user privacy protection. When aggregating local model parameters into a GM, differential privacy can be employed to introduce noise on model parameters, protecting the details of individual models. This ensures that the GM's training does not overly rely on the specific data of any one participant. Differential privacy techniques can be applied to gradient computation and updates, introducing noise on gradients to protect individual data (Dwork 2011).

5.4 Data Compression

Compression solutions involve employing various techniques to reduce or compress the amount of data transmitted in FL. This helps to lower communication overhead, improve the efficiency of model updates, and maintain the accuracy of model training. When applying differential privacy, the size of transmitted noisy data can be reduced by adjusting the parameters of the noise or using more efficient differential privacy algorithms. Sparse ternary compression (STC) can significantly reduce the model size, thereby lowering memory and computational costs when deploying on embedded or edge devices. The active participation of numerous clients also ensures the robustness of the model (Zhou et al. 2021, Sattler et al. 2019).

6 PRIVACY PROTECTION ALGORITHM COMPARATIVE ANALYSIS

6.1 Siren

Siren, a Byzantine-robust FL system with an active alert mechanism, improves defense against attacks by employing precision checks and distributed detection. Each client conducts two processes: training and alert. In the training process, a small portion of the local dataset is retained as a test dataset. The alert process tests the global weights, and alerts are sent to the CS to remove malicious weights during each communication round (Guo et al. 2021).

6.2 Edge Computing Privacy Protection

This system utilizes blockchain for decentralization and auditability, bolstering resistance to tampering and single-point failure attacks. FL establishes a collaborative training platform across multiple devices without requiring a trusted environment or specialized hardware. It incorporates adaptive differential privacy to protect model parameter privacy while reducing noise's impact on model accuracy. This integration offers a solution with high accuracy and robust privacy protection for edge computing scenarios (Fang et al. 2021).

6.3 FLAME

The FLAME framework combines differential privacy and FL, achieving the goal of simultaneously protecting user privacy and improving model accuracy without requiring a trusted party, using the shuffling model in differential privacy. It balances model accuracy and user privacy protection, avoids some limitations of traditional models, and offers better performance for practical applications. It also demonstrates strong resistance against poisoning attacks (Liu et al. 2021).

6.4 Summary

Table 2 summarizes different architectures for protection against attacks, highlighting their methods, advantages, disadvantages, and defense mechanisms.

Table 2. Privacy protection algorithm comparison

Architecture	Protection Methods	Advantages	Disadvantages	Defense Against Attacks
Siren	Distributed Detection	Suitable for a large number of malicious clients	No Apparent Drawbacks	Various Attacks
Edge Computing Privacy Protection	Adaptive Differential Privacy Mechanism, Gradient Checking, and Incentive Mechanism	Suitable for scenarios with high security and accuracy requirements.	Low efficiency	Poisoning Attacks
FLAME	Privacy amplification benefit	Performance improvement, avoiding limitations	Not suitable for large parameter dimensions	Vulnerable to poisoning attacks

7 DISCUSSION AND ANALYSIS

By learning models in a distributed environment, model training can be achieved without centralizing data. Communication efficiency is crucial, especially when learning on mobile devices and reducing communication rounds is vital for performance improvement. Different technologies and methods, such as iterative model averaging, model accuracy checks, and model alert mechanisms, can be employed. Future research could explore the applicability of these methods in broader and more complex scenarios, as well as how to enhance model robustness and privacy protection performance further.

In the field of FL, there is a need for more attention to comprehensive optimization methods that address communication efficiency, security, and model performance simultaneously.

8 CONCLUSION

In the field of FL, the technology to address the issue of data silos has garnered significant attention. Despite having certain privacy protection mechanisms, FL still poses risks of privacy leakage, especially in sectors such as healthcare and finance, where the demand for user privacy protection is urgent. The paper reviews the fundamental principles, classifications, and privacy challenges of FL, with a particular focus on privacy threats like Byzantine attacks, poisoning attacks, and Sybil attacks.

Regarding privacy protection, researchers have proposed various methods, including homomorphic encryption, differential privacy, and data compression technologies. Homomorphic encryption enables computational operations on encrypted data, effectively safeguarding the privacy of model parameters and input data. Differential privacy protects data privacy on local devices by introducing noise and prevents overreliance on individual models by introducing noise on model parameters. Data compression technology enhances communication efficiency by reducing the amount of transmitted data while maintaining the accuracy of model training.

In the comparative analysis of privacy protection algorithms, Siren employs an active alert mechanism, edge computing privacy protection combines blockchain technology, and the FLAME framework integrates differential privacy with FL. These methods not only enhance model accuracy but also effectively counter various types of privacy attacks.

Overall, as a distributed machine learning approach, FL faces challenges in the comprehensive optimization of communication efficiency, security, and model performance. Future research should delve into the applicability of these methods in broader and more complex scenarios to further enhance the robustness and privacy protection performance of FL.

REFERENCES

A. N. Bhagoji, S. Chakraborty, P. Mittal, et al. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, (2019), pp. 634-643.

C. Dwork. Communications of the ACM, 54(1), 86-95, (2011).

C. Fang, Y. Zheng, Y. Wang, et al. Journal of Communications, 42(11), 28-40, (2021).

C. Zhou, Y. Sun, D. Wang, et al. Journal of Network and Information Security, 7(5), 77-92, (2021).

D. A. E. Acar, Y. Zhao, R. M. Navarro, et al. arXiv preprint arXiv:2111.04263, (2021).

E. Bagdasaryan, A. Veit, Y. Hua, et al. “How to backdoor federated learning”. In *International conference on artificial intelligence and statistics*, (2020), pp. 2938-2948.

F. Sattler, S. Wiedemann, K. R. Müller, et al. IEEE transactions on neural networks and learning systems, 31(9), 3400-3413, (2019).

H. Guo, H. Wang, T. Son, et al. “Siren: Byzantine-robust federated learning via proactive alarming”. In *Proceedings of the ACM Symposium on Cloud Computing*, (2021), pp. 47-60.

H. Wang, M. Yurochkin, Y. Sun, et al. arXiv preprint arXiv:2002.06440, (2020).

IEEE Computer Society. “IEEE Guide for Architectural Framework and Application of Federated Machine Learning.” in *IEEE Std 3652.1-2020*, (2021), pp.1-6.

J. Chen, J. Chu, M. Su, et al. Journal of Information Security, *5*(4), 14-29, (2020).

J. Konečný, H. B. McMahan, D. Ramage, et al. arXiv preprint arXiv:1610.02527, (2016).

J. Wang, L. Kong, Z. Huang, et al. Big Data, 7(3), 130-149, (2021).

J. Xu, B. S. Glicksberg, C. Su, et al. Journal of Healthcare Informatics Research, 5, 1-19, (2021).

J. Zhang, S. Guo, X. Ma, et al. Advances in Neural Information Processing Systems, 34, 10092-10104, (2021).

M. Kang, J. Wang, D. Li, et al. Chinese Journal of Intelligent Science & Technology, 4(2), (2022).

N. Baracaldo, B. Chen, H. Ludwig, et al. “Mitigating poisoning attacks on machine learning models: A data provenance based approach”. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, (2017), pp. 103-110.

N. Baracaldo, B. Chen, H. Ludwig, et al. “Mitigating poisoning attacks on machine learning models: A data provenance based approach". In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, (2017), pp. 103-110.

Q. Li, B. He, D. Song. "Model-contrastive federated learning". In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (2021), pp. 10713-10722.

Q. Yang, Y. Liu, T. Chen, et al. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19, (2019).

R. Liu, Y. Cao, H. Chen, et al. "Flame: Differentially private federated learning in the shuffle model". In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10), (2021), pp. 8688-8696.

Tensorflow Federated. TensorFlow, 2024, available at https://www.tensorflow.org/federated?hl=zh-cn

W. Jiang, H. Li, S. Liu, et al. A flexible poisoning attack against machine learning. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, (2019), pp. 1-6.