Anti-poisoning federated learning with differential privacy

Chenxi Huang

25 Nov, 2021

Abstract

Cross-platform knowledge sharing strongly desires secure data analysis decentralization and cooperation. To address the trade-off problem between privacy and utility, defending poisoning attack and model inversion in complex federated learning, this research firstly focuses on statistically quantifying the inherent indistinguishability between models, based on hypothesis testing. Secondly, a filtering mechanism for detecting malicious adversaries is generated. Thirdly, honest participants dynamically adjust their protection level. Finally, this research intends to design and implement a novel anti-poisoning federated learning framework with differential privacy.

Keywords

Federated learning, Differential Privacy, Poisoning attack

1 Introduction

Nowadays, artificial intelligence faces two main challenges: data silos and privacy concerns [1, 2]. Meanwhile, with the widely used edge and Internet of things devices, the federated learning framework, as shown in Figure 1, attacks extensive attentions, which can train the global model based on the mode of decentralization and cooperation. However, there are still severe privacy protection problems in federated learning, although models are trained by sharing updates (such as gradient information) rather than raw data [3, 4]. Recent studies show that, by analyzing the parameter differences between training and uploading, privacy can still be leaked to a certain extent, such as the weight of neural network training [5, 6]. Federated learning requires different participants to upload and aggregate parameters repeatedly to train the global model. This process leads to personal privacy disclosure through the model-inversion attack.

Differential privacy (DP) provides a strict, quantifiable, and context-independent privacy protection method for machine learning. For its information theory guarantee, DP is also widely used to enhance data privacy with its simplicity and low cost [7, 8, 9, 10]. However, as the fundamental challenge of privacy-preserved methods, DP mechanisms inevitably cause model performance degradation and system utility loss. Traditional DP injects bounded noise into model to protect privacy, and privacy budget is a crucial factor in measuring the level of protection and the amount of noise. Existing differentially private methods allocate the same amount of privacy budget for each participant and each iteration in model updating, which leads to ubiquitous trade-off problems between privacy and utility [7, 9, 10].

Federated learning, on the other hand, is inherently vulnerable to poisoning attacks because local training samples are not released to trusted curator for inspection. If machine learning models are trained based on data from potentially unreliable sources, attackers can easily reduce model performance by inserting elaborate malicious samples into the training set. This kind of manipulation is known as poisoning attack [11]. Many studies explore the threat of poisoning attacks in federated learning and propose countermeasures [12, 13, 14]. However, few studies have investigated the poisoning attacks of federated systems in which high-dimensional datasets and differential privacy protection involved. It is unclear how the high dimensionality of datasets affects the performance and time efficiency of poison attack and its defense measures.

DP-based federated learning systems, while privacy can be protected by adding noise, come at the expense of learning accuracy, and are inherently vulnerable to poisoning attacks. Therefore,

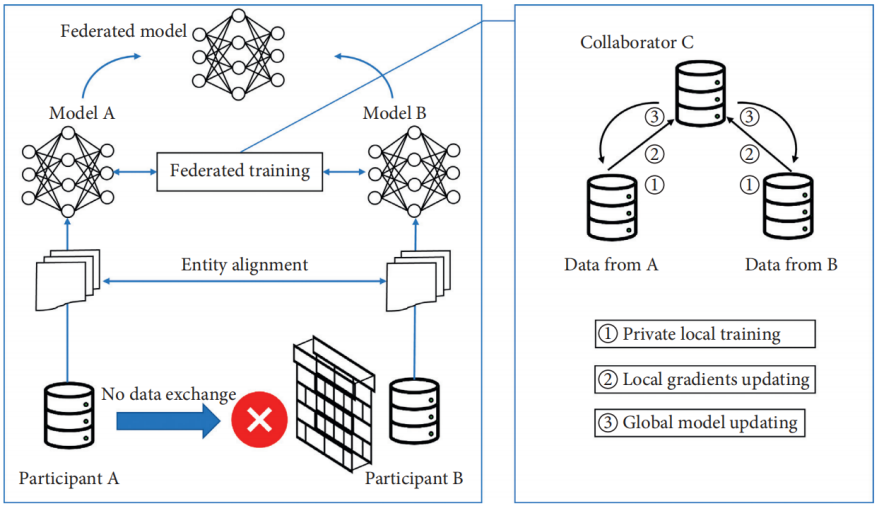

Figure 1: Framework of classical federated leaning

designing and implementing participant filter mechanism, conducting dynamic privacy budget allocation, and aggregating cleaner data sets into servers are effective means to improve the effectiveness of federal system and resist poison attack. This research intends to study privacy protection methods in federated learning, aiming to achieve higher accuracy under the same noise level, and resist poisoning attacks more effectively in federated learning systems with high-dimensional data.

2 Literature Review

A series of theoretical and methodological research on privacy-preserved federated learning has been conducted, and many research results have been obtained. Privacy in federated learning can be divided into global differential privacy(GDP) [15] and local differential privacy (LDP) [7]. Global privacy requires that the model updates generated in each round be privacy-protected for all untrusted third parties except the central server. In contrast, local privacy further requires that the update be privacy-protected for the server. The privacy protected method research in federated learning continues and extends in traditional machine learning, which are mainly based on multiparty secure computing [16, 17] and differential privacy [7, 18, 19]. Among them, multiparty secure computing is a kind of the lossless method, which can maintain the original accuracy and make a strong privacy guarantee. However, the approach results significant additional communication costs. Considering the high communication cost in the federated system, differential privacy has low system overhead. Existing studies include federated learning algorithms that satisfy LDP [7], differentially private stochastic gradient descent algorithm (DP-SGD) [18], and metalearning with DP [20].

Many works have been done to explore the threat of poisoning attacks in federated learning and propose countermeasures. In single attacker scenario, an adversariy easily carry out model poisoning attacks while maintaining stealth [12]. Adversarial training can defend against such attacks by preventing the model from learning trends specific to individual parties data [13]. In multiple attackers scenarios, systems identify poisoning sybils based on the diversity of client updates in the distributed learning process [21]. System *Auror* detects malicious users, generates an accurate model, and achieves a only 3% accuracy drop even when 30% of all the users are adversarial [22]. After observing the relations between the number of poisoned training samples, attackers, and attack success rate, a scheme, Sniper, eliminates poisoned local models from malicious participants during training [14].

3 Existing Problems

As the fundamental challenge of privacy-preserved methods, DP inevitably causes utility loss. Focusing on the trade-off between privacy and utility in the model training process, many state-

of-the-art differentially private federated algorithms have been proposed [23, 24, 25, 26, 27]. There are still several issues of the method design.

- These methods only consider the privacy budget allocation for different participants, but they do not consider the privacy budget cost on the process of the global model iteration.
- Few studies focused on the impact of differential privacy protection on the poisoning attacks and their defenses, considering noise intake before uploading is bound to change the models themselves.
- Poisoning attacks lack evaluation on perfomance and efficiency regarding high dimensionality, sparseness, and correlation of datasets. Consequently, the defense optimization of poisoning attack against these characteristics of the dataset is lacking.
- Differential privacy provides limited protection against poisoning attack, though performing well in defending model inversion.

4 Objectives

In view of the above problems, this project intends to study the following objectives.

- Study the privacy protection methods in federated learning, explore how DP method affects poisoning attacks, investigate the influence of high dimensionality, sparseness and record correlation of data on poisoning attack and defense efficiency
- Based on the threat models of poisoning attack and model inversion, formalize four utility metrics of accuracy, convergence, time utility and robustness of complex federated systems comprehensively
- Quantify the inherent indistinguishability between models statistically, generate a clients filtering mechanism, dynamic privacy budget allocation and training, aggregate a cleaner dataset into the server, to achieve higher accuracy at the same noise level
- Analyze how the privacy budget allocation methods affect the convergence performance. Based on theoretical research, design the privacy budget dynamic allocation method, reduce the noise intake to improve utility.
- Plan to design and implement an anti-poisoning FL framework with DP. The framework counter the two threats above, while comprehensively improving the utility of four metrics, to achieve the trade-off of privacy and utility.

5 Methodology

The research method generally includes three aspects.

1. On the whole, the research is planned to be carried out from six aspects
   - Formalization of Federated Learning
   - Local model upload
   - Participant distribution distance analysis and evaluation
   - Formalization of Threat model
   - Design of Participant filter and privacy budget allocation
   - Global AP-DP model design and verification
2. Technically, use the differential privacy protection theory and technology, federated learning and optimization theory and technology, as well as deep learning theory and related technical methods to carry out relevant theoretical analysis, combine algorithm design and empirical research with the actual background of poisoning attack technology
3. Academically, actively carry out international academic exchanges. Attend top international conferences on privacy protection, artificial intelligence and digital machine learning (ICDE, IDSC, IKDE, etc.) to master international research frontiers in privacy protection methods and technologies, machine learning, federated learning and Internet of Things, etc.

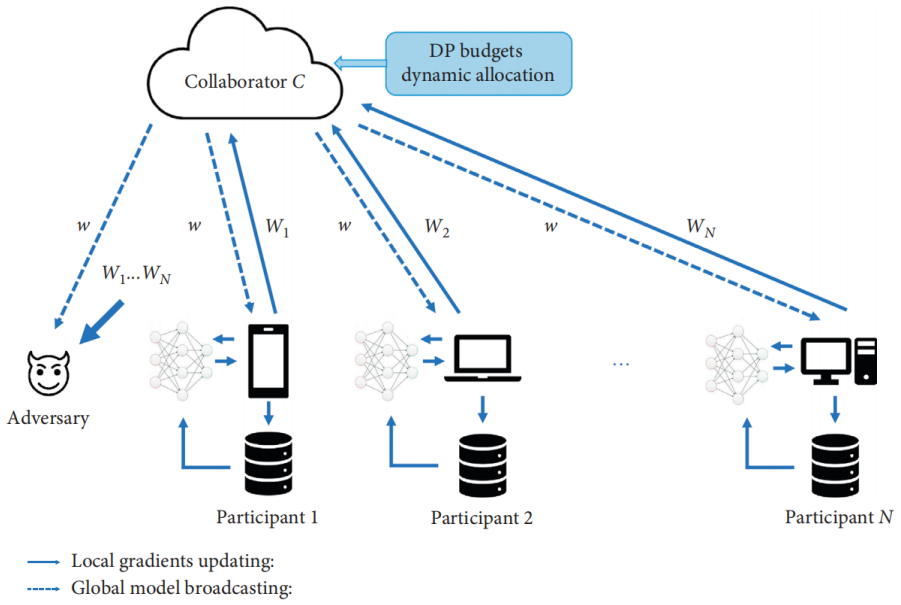

Figure 2: DP budget dynamic allocation FL framework

5.1 Formalization of Federated Learning

A federated learning system, as shown in Figure 2, trains the global model by introducing different samples from local datasets into the common feature space, namely, $X_i = X_j, Y_i = Y_j$, and $\forall D_i, D_j, i \neq j$. The system consists of a trusted collaborator C and N participants. D_i represents local dataset of participant i, $i \in 1, 2, ..., N$. Defining $f_i : R^d \rightarrow R$ as the loss function of participant i, federated learning can be formalized as the following optimization problem:

$$\min_{\omega \in R^d} f(\omega) := \frac{1}{N} \sum_{i=1}^{N} f_i(\omega) \tag{1}$$

In order to optimize problem 1, the federated learning process of training a global model is designed as follows:

1. Local model training: the participants conduct a round of parameters' training according to the local data and upload the local parameters of the current round to the collaborator
2. Secure aggregating: the collaborator performs secure aggregation for the current round of local parameters uploaded by the participants
3. Global parameter broadcasting: the collaborator broadcasts the aggregation parameters of the current round to the participants
4. Local model updating: the participants update the model according to the global parameters of the current round and conduct the next round of training

5.2 local model upload

Each participant trains their local model and perturbs parameters for uploading safety. In this phase, participants utilize DP-SGD [23] and Gaussian mechanism in DP. DP-SGD adjusts the model weights ω to minimize the error function. Gaussian noise is added to the gradient calculated by the data in the current batch, and then, the average gradient of the current batch is calculated:

$$\widetilde{g}_t \leftarrow \frac{1}{B}(\sum_i g_t(x_i) + N(0, \sigma_t^2 C^2)) \tag{2}$$

where B is the number of current batch of data. $g_t(x_i)$ represents the gradient of the loss function calculated by training data x_i to the weight ω in the tth iteration. $N(0, \sigma_t^2 C^2)$ represents

Gaussian noise with mean 0 and variance $\sigma_t^2 C^2$. σ is the key parameter to control the noise scale. The greater the σ, the greater the variance of the normal distribution and the greater the amount of Gaussian noise.

In order to prevent gradient explosion that results in poor model convergence performance, Gradient is clipped from $\widetilde{g}_t$ to $\hat{g}_t$. Gradient clipping ensures that the second norm of the gradient does not exceed the clipping threshold C. Following clipping, the model update according to the learning rate η to obtain the local parameters.

$$\widetilde{\omega}_i^{(t)} \leftarrow \omega_i^{(t)} - \eta \hat{g}_t \tag{3}$$

5.3 Participant indistinguishability

Most defense mechanisms against poison attacks rely on the fact that poison samples are usually outside the expected input distribution. Thus, poisoned samples can be treated as outliers and training samples can be purified using data cleansing (i.e., attack detection and deletion). In other words, in order to force the aggregated global model deviating from the benign model, the poisoned local models have to be different from those benign local models trained by honest participants. Euclidean distance, the most common ways to measure distance in high dimensional space, is used to demonstrate the difference between two local models M_1, M_2 [14].

$$Dis(M_1, M_2) = \sqrt{\sum (M_1 - M_2)^2} \tag{4}$$

However, Euclidean distance assumes that each feature in the dataset contributes equally. This evaluation sometimes fails to meet practical requirements. Moreover, it does not take into account the data sparseness and record correlation. Naturally, the harder the two models are to distinguish, the closer they are to each other. Thus, inspired by the indistinguishability for two different distributions [28], the notion of indistinguishability can be formalized as a hypothesis testing problem. For two datasets S_1 and S_2, set two simple hypotheses and two according types of errors.

$$H_0\text{: the true dataset is } S_1$$

$$H_1\text{: the true dataset is } S_2$$

$$\text{type I error: the probability of erroneously rejecting } H_0 \text{ when } H_0 \text{ is true}$$

$$\text{type II error: the probability of erroneously accepting } H_0 \text{ when } H_1 \text{ is true}$$

the optimal trade-off between the type I and type II errors delineate the difficulty in distinguishing the two hypotheses. Consider a rejection rule $0 \leq \phi \leq 1$, with type I and type II error rates defined as

$$\alpha_\phi = E_{M(S_1)}[\phi] \tag{5}$$

$$\beta_\phi = 1 - E_{M(S_2)}[\phi] \tag{6}$$

respectively, where M denotes a randomized algorithm. Fixing the type I error at any level, evaluating the minimal achievable type II error, motivate the following *indistinguishability* definition.

$$\begin{aligned} ind(M(S_1), M(S_2)) : [0,1] &\mapsto [0,1] \\ \alpha &\mapsto \inf_\phi \{\beta_\phi : \alpha_\phi \leq \alpha\} \end{aligned} \tag{7}$$

Writing $f = ind(M(S_1), M(S_2))$, the definition says that $f(\alpha)$ is the minimum type II error among all tests at significance level α.

5.4 Threat model

The threat in this research comes from malicious participants and server. Assuming there are multiple malicious attackers, all of them have the same attack goal. That is, all attackers aim to mislead the global model to the same direction. Attackers use label flipping to generate the poisoned sample, that is, changing the label of the training sample and keeping the sample characteristics unchanged. Label flipping does not require any knowledge of classifiers and pre-training. The

number of attackers should not exceed one-third of the total number of participants. From an attack's perspective, it succeeds if the poisoned model outputs the desired target label T instead of a source label I. Therefore, the success rate SR of a global model satisfies

$$SR_M = \frac{n_T^{(I)}}{n^{(I)}} \times 100 \tag{8}$$

where $n^{(I)}$ indicates the number of testing samples with the source label I, and $n_T^{(I)}$ indicates the number of testing samples mislabeled as T.

5.5 Participant filter

At each iteration, before aggregating local models, the server needs to rule out malicious local models to avoid obtaining contaminated global models. The detailed filtering process of the server is as follows.

1. The server collects locally noised model M_i from all N participants, calculates the *indistinguishability* $ind(M_i, M_j)$ between each pair of local models
2. The server treats each local model as a vertex. If the indistinguishability between two models (vertices) is less than the threshold θ, there is an edge between the two models.
3. the server finds the maximum clique. If the number of vertices in the clique is greater than N / 2, the privacy budget is reallocated to the vertices in the clique for aggregation, otherwise increase θ with a fixed step size and go to step 2.

5.6 Noise scale adjustment

Theoretically, when all global participants N participate in training, the convergence performance of the federated system is related to the number of iterations T and the noise level. As T increases, the convergence performance improves, but the accumulated noise level also increases, which leads the convergence performance decrease. A mitigating way, is to reduce noise scale dynamically by verifying the accuracy of the current global model, under a certain global privacy level. Another way is to adjust noise scale dynamically in local training by verifying the accuracy change between two rounds. Every time the increase in local accuracy is lower than the threshold, the noise scale is reduced by k times, until the total privacy budget is exhausted. The first way requires the monitering from the collaborator, while the second is more like an introspection. When the local model M_i accuracy improvement $S_t^{(i)} - S_{t-1}^{(i)}$ between round t and round $t-1$ is less than the threshold α, the participant will reduce the noise parameter to $\sigma_t^{(i)'} = k\sigma(i)_t$ $(k \in (0,1))$, otherwise the noise scale remains unchanged. Subsequently, the updated noise scale will be used in the subsequent training until next noise scale adjustment happens.

$$\sigma_t^{(i)'} = \begin{cases} k\sigma_t^{(i)}, & S_t^{(i)} - S_{t-1}^{(i)} \leq \alpha \\ \sigma_t^{(i)} \end{cases} \tag{9}$$

where $k \in (0,1)$ and $S_0 = 0$. $\sigma_t^{(i)}$ indicates the current noise scale of local model M_i in tth iteration.

5.7 Global AP-DP model

According to the threat defense and solving the optimization problem formalized mentioned above, this research intends to design an anti-poisoning and differentially private federated learning framework(AP-DP), as shown in Figure 3. For each iteration, it works as follows.

1. *Local model upload.* Each participant trains their local models, adds noise to perturb weights of the models, uploads models to the collaborator
2. *Filter.* The collaborator uses filtering mechanism in Section 5.5 to filter out the malicious participants
3. *Aggregation.* The collaborator aggregates the honest participants as a new global model

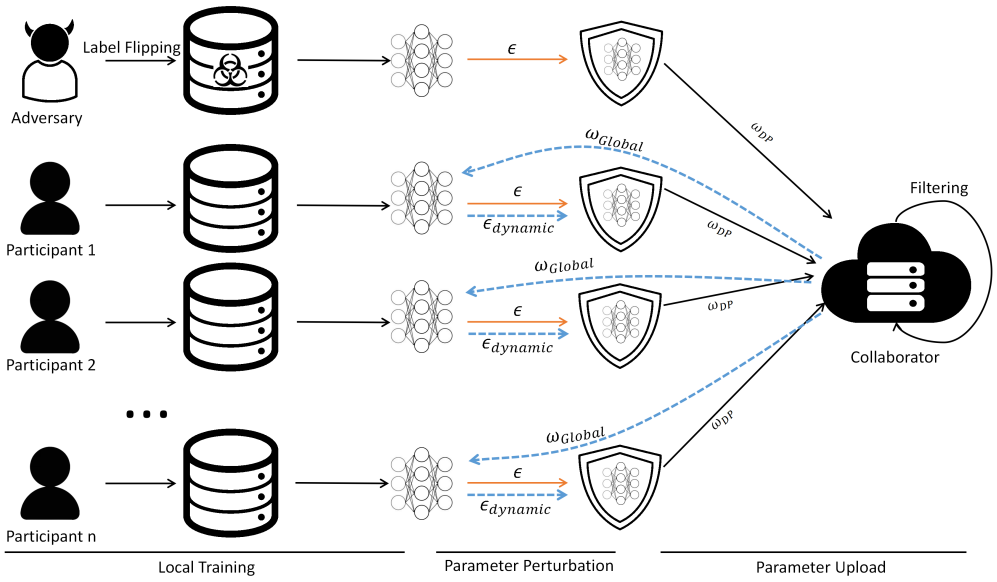

Figure 3: AP-DP federated learning framework

4. *Noise scale adjustment.* The participants verify their own local model convergence situation and decide whether adjusting the noise parameter or not in the current predetermined dynamic adjustment round. If so, an adjusted noise parameter will be utilized in the next training round

This framework satisfies global differential privacy for the federated system and optimizes the utility by dynamic allocating differential privacy budgets. By injecting Gaussian noise to local parameters, the designed federated system achieves (ϵ, δ)-differential privacy. Furthermore, the server identifies potentially malicious data sets through *indistinguishability* evaluations, a kind of distance between distributions based on hypothesis testing. Filtered out the identified adversaries, the collaborator conducts the privacy budgets dynamic allocation according to the utility performance of honest participants and iteration process.

6 Experiment Means

In order to verify the effect of the method of this research on privacy protection, it is planned to follow the steps, algorithm design and implementation → experimental design and environment construction → verification and implementation, to conduct experimental verification of the proposed method.

1. *Platforms.* Develop proposed algorithms based on Python and Tensorflow platform, simulate the relevant designed algorithms.

2. *Datasets.* Choose MNIST and CIFAR-10 datasets for comparison. Use label flipping to generate the poisoned sample, that is, changing the label of the training sample and keeping the sample characteristics unchanged, because it does not require any knowledge of classifiers and pre-training

3. *Baselines.* Select *Sniper* and *Auror* as the comparison algorithms

4. *Metrics.* Add equal noise to the comparison algorithms to achieve global differential privacy protection, set the equal number of poisoned samples in comparison algorithms, and compare the accuracy, convergence performance, time utility and success rate of poison attack of different algorithms to verify the effectiveness of the algorithm

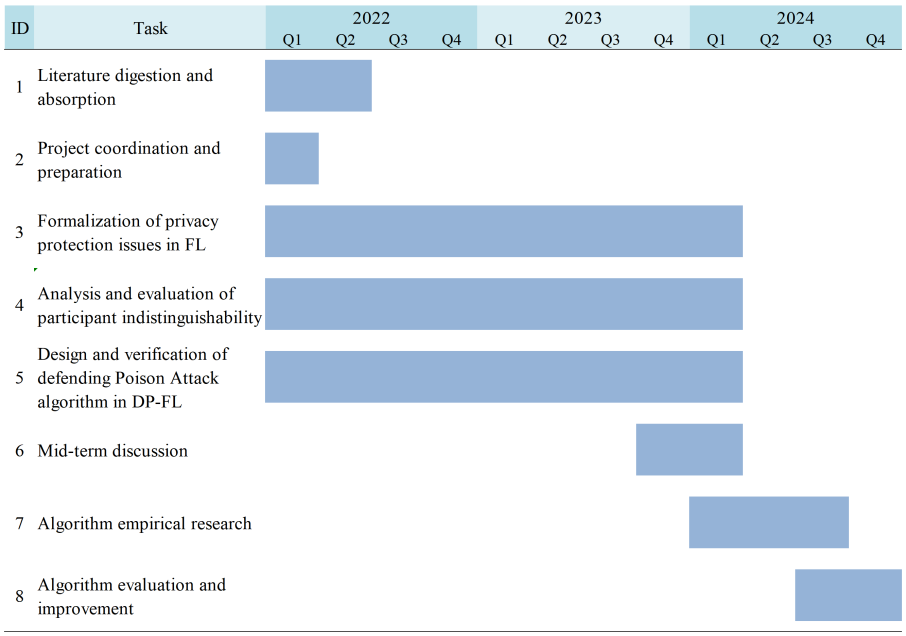

Figure 4: Annual schedule

7 Schedule

The overall annual schedule of the research is as follows, and the specific research schedule is shown in Figure 4.

2022.1-2022.12

1. Summarize and learn federated system theory and technical methods
2. Summarize and study existing federal learning research on differential privacy protection
3. Study and analyze optimization problems in federated system
4. Study and analyze the privacy protection in federal learning under different scenarios

2023.1-2023.12

1. Study and define the poisoning threat of participants in the federal system, study the analysis and evaluation methods of distance between models and between distributions
2. Study the problem of poisoning attack in federal system, and study the analysis and evaluation method of poisoning attack defense means
3. Study the dynamic allocation method of privacy budget for global iterative process
4. Study the dynamic allocation method of privacy budget of selected participants in a round of iteration

2024.1-2024.12

1. Integrate research results, design and implement anti-poisoning and differentially private federated learning framework(AP-DP)
2. Empirical research on algorithms in AP-DP
3. Improve algorithms in AP-DP
4. Summarize research results and conduct project review

8 Conclusion

This research focuses on the study of privacy and utility balance problems in federated learning system. The specific innovations include:

1. Based on the threat models of poisoning attack and model inversion, investigate four utility metrics of accuracy, convergence, time utility, and robustness of complex federated systems, so as to formalize the utility optimization problem more comprehensively.

2. To better adapt to datasets' characteristics, attempt to quantify the inherent indistinguishability between models statistically, which helps to generate a filtering mechanism to defend poisoning. Study the poisoning attack and defense efficiency on datasets' characteristics.

3. Analyze how the privacy budget allocation methods affect the convergence performance. Based on theoretical research, design the privacy budget dynamic allocation method, therefore reduce the noise intake to improve utility.

4. Plan to design and implement an anti-poisoning FL framework with DP. The framework counters the two threats above, while comprehensively improving the utility of four aspects, so as to achieve the trade-off of privacy and utility.

References

[1] Y. Cheng, Y. Liu, T. Chen, and Q. Yang. Federated learning for privacy-preserving AI. *Communications of the ACM*, 63(12):33–36, 2020.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12–19, 2019.

[3] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[5] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[6] L. Melis, C. Song, and E. De Cristofaro. Exploiting unintended feature leakage in collaborative learning. *Proceedings of the IEEE Symposium on Security Privacy*, page 691–706, 2019.

[7] N. Wang. Collecting and analyzing multidimensional data with local differential privacy. *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE)*, page 638–649, 2019.

[8] Q. Yang. *Federated Learning*. Morgan Claypool, 2019.

[9] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. *Proceedings of the International Conference on Learning Representations*, 2018.

[10] G. Andrew, O. Thakkar, and H. B. McMahan. Differentially private learning with adaptive clipping. 2019, http://arxiv.org/abs/1905.03871.

[11] Benjamin I. P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. ANTIDOTE: understanding and defending against poisoning of anomaly detectors. *Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference*, pages 1–14, 2009.

[12] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. *International Conference on Machine Learning*, pages 634–643, 2019.

[13] J. Hayes and O. Ohrimenko. Contamination attacks and mitigation in multi-party machine learning. *Advances in Neural Information Processing Systems*, pages 6604–6615, 2018.

[14] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun. Understanding Distributed Poisoning Attack in Federated Learning. *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 233–239, 2019.

[15] K. Wei, J. Li, and M. Ding et al. Federated learning with differential privacy: algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[16] K. Bonawitz, V. Ivanov, and B. Kreuter et al. Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the Conference on Computer and Communications Security*, 2017.

[17] B. Ghazi, R. Pagh, and A. Velingker. Scalable and differentially private distributed aggregation in the shuffled model. 2019, http://arxiv.org/abs/1906.08320.

[18] N. Wu, F. Farokhi, D. Smith, and M. Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. *Proceedings of the IEEE Symposium on Security and Privacy*, page 304–317, 2020.

[19] J. Li, M. Khodak, S. Caldas, and A. Talwalkar. Differentially private meta-learning. *Proceedings of the International Conference on Learning Representations*, 2020.

[20] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in theoretical Computer Science*, 9(3-4):211–407, 2013.

[21] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. Mitigating Sybils in Federated Learning Poisoning. 2020, https://arxiv.org/abs/1808.04866.

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

[23] J. Zhang, Y. Zhao, J. Wang, and B. Chen. FedMEC: improving efficiency of differentially private federated learning via mobile edge computing. *Mobile Networks and Applications*, 25(6):2421–2433, 2020.

[24] K. Wei, J. Li, and M. Ding et al. Federated learning with differential privacy: algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[25] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 7(10):9530–9539, 2020.

[26] X. Liu, H. Li, G. Xu, R. Lu, and M. He. Adaptive privacy-preserving federated learning. *Peer-to-Peer Networking and Applications*, 13(6):2356–2366, 2020.

[27] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Transactions on Industrial Informatics*, 16(10):6532–6542, 2020.

[28] Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian Differential Privacy. 2019, https://arxiv.org/abs/1905.02383.