Differential Privacy Federated Learning: A Comprehensive Review

Fangfang Shan[1*], Shiqi Mao[2], Yanlong Lu[3], Shuaifeng Li[4]
School of Computer Science, Zhongyuan University of Technology, Zhengzhou 450007, Henan, China[1, 2, 3, 4]
Henan Key Laboratory of Cyberspace Situation Awareness, Zhengzhou 450001, Henan, China[1]

***Abstract*—Federated Learning (FL) has received a lot of attention lately when it comes to protecting data privacy, especially in industries with sensitive data like healthcare, banking, and the Internet of Things (IoT). However, although FL protects privacy by not sharing raw data, the information transfer during its model update process can still potentially leak user privacy. Differential Privacy (DP), as an advanced privacy protection technology, introduces random noise during data queries or model updates, further enhancing the privacy protection capability of Federated Learning. This paper delves into the theory, technology, development, and future research recommendations of Differential Privacy Federated Learning (DP-FL). Firstly, the article introduces the basic concepts of Federated Learning, including synchronous and asynchronous optimization algorithms, and explains the fundamentals of Differential Privacy, including centralized and local DP mechanisms. Then, the paper discusses in detail the application of DP in Federated Learning under different gradient clipping strategies, including fixed clipping and adaptive clipping methods, and explores the application of user-level and sample-level DP in Federated Learning. Finally, the paper discusses future research directions for DP-FL, emphasizing advancements in asynchronous DP-FL and personalized DP-FL.**

***Keywords*—Federated learning; differential privacy; privacy protection; gradient clipping**

I. Introduction

Concerns over data security and privacy, particularly in the context of the Internet of Things (IoT), healthcare, and finance, have led to a surge in the adoption of federated learning (FL) technologies in recent years. For instance, FL can be used for disease monitoring [1], financial analysis [2], and IoT data sharing [3] FL enables the training of models using data from multiple participants without sharing sensitive data, thus obtaining a broader and more representative data perspective. Despite its significant advantages in protecting data privacy, FL involves the transmission and sharing of data and model parameters between participants, which raises concerns about the security and integrity of communications. If the communication channels are not protected or are vulnerable to man-in-the-middle attacks, it can lead to issues such as data leakage, tampering, or forgery. In fact, even if attackers cannot directly access the datasets, user privacy can still be threatened. By analyzing model parameter updates, attackers can infer information about the original data [4], a type of attack known as an inference attack. Attackers can also introduce false labels or tags into the training set, a technique called as "data poisoning" [5], which lowers the accuracy of the model's predictions by making it learn the wrong patterns. It is vital to safeguard privacy and fend against these attacks in FL as a result.

In this paper, we introduce various techniques proposed to address privacy issues in Federated Learning. Specifically, we focus on Differential Privacy (DP), which has become the de facto standard for protecting user privacy in statistical computations. These techniques can be categorized into three types:

- Data Privacy Protection: The goal is to protect raw data from being leaked or illegally accessed. Key techniques include Differential Privacy, which reduces the risk of data identification by adding random noise to data queries or statistical processes; Data privacy is preserved during computations thanks to homomorphic encryption, which enables calculations on secret information without the need to decrypt it; and Data Masking techniques, which prevent identification by altering the structure or form of the data.
- Model Privacy Protection: This aims to protect trained models from reverse engineering or illegal analysis. Techniques include Model Compression and Model Distillation. Model Compression reduces the complexity and number of parameters in a model, thereby lowering the risk of model leakage. Model Distillation involves transferring the knowledge of a large model to a smaller, simpler model, reducing the amount of data that needs protection. Additionally, Model Watermarking techniques embed specific markers in the model to track and protect its usage.
- Communication Privacy Protection: This focuses on securing data transmission during the communication process in Federated Learning. To guarantee the safety and confidentiality of data during transmission, it mainly uses secure communication protocols and encryption technologies, such as Secure Sockets Layer/Transport Layer Security (SSL/TLS). Additionally, Trusted Execution Environments (TEEs) allow for secure data aggregation and model updates without revealing individual inputs.

1) Data privacy protection: Secure Multi-Party Computation (SMC) [6][7] enables multiple parties (also known as entities) to collaboratively compute any function on secret data without revealing any other secret information besides the function's output. The concept of SMC was introduced by the academic community in the 1980s, along with various feasible design methods for MPC protocols for any function. These design methods form the basic framework for most subsequent MPC protocols. SMC ensures that the inputs are neither disclosed to each other nor to a central server, thus doing away with the requirement for a reliable third party.

A unique type of encryption called homomorphic encryption (HE) [8] [9] permits certain actions to be carried out on encrypted data while maintaining the data's encryption. The plaintext and the outcome of applying the identical procedures directly to the original plaintext data are consistent when the encrypted result is decrypted. HE has now evolved to support floating-point operations with the fourth generation FHE schemes. The primary feature of the fourth generation FHE schemes is their support for floating-point homomorphic operations. In 2017, Cheon et al. introduced the CKKS scheme in ASIACRYPT 2017, which for the first time handled floating-point numbers in FHE [10]. Although the CKKS scheme is simpler and offers improved performance, making it a strong privacy protection tool, its computational complexity makes using HE in FL practically inefficient, particularly in cases where the training dataset exceeds the capacity of the computer's memory.

Differential Privacy (DP) [11][12] is an advanced privacy protection technique that allows for the analysis and release of datasets without compromising individual privacy. DP accomplishes this by balancing the trade-off between data utility and individual privacy by injecting controlled noise into the data analysis process. Although the first definition of DP appeared in 2006, it has only recently gained attention for practical applications. Accuracy is the primary obstacle to the practical application of DP; accuracy is frequently diminished when privacy protection is increased. Investigators attempt to resolve this issue by integrating DP with other techniques to ensure its usability or by attempting to reconcile privacy and accuracy.

2) Model privacy protection: Knowledge Distillation (KD) [13][14][15] does not transmit model updates but instead, if the local model size is greater than the public dataset, communicates local model predictions among several clients on a shared public dataset, saving communication costs. In its initial form, information is passed on by simulating the output of the teacher model on the same set of data. Subsequent research revealed the function copying might guide student model training in addition to imitating outcomes [16]. These days, Federated Learning (FL) frequently uses KD as a standard technique [18][19]. It is possible to apply alternative solutions in an adaptable manner to different scenarios while still imitating the global model and the local preceding model. In order to minimize shared bits, Li and Wang [14] investigated Federated Knowledge Distillation by averaging logits for each sample. Gong et al. To solve telecommunication inefficiencies, [16] suggested a one-shot learning paradigm for one-way distillation. Knowledge was extracted from anticipated soft labels and subsequent results by Wu et al. [17]. Compared to device selection-based and model compression-based approaches, KD-based systems share fewer bits in each interaction cycle and do not require a trade-off between model accuracy and the number of participating devices. While significantly reducing communication overhead.

Model Watermarking is a technique used to protect the intellectual property of deep learning models. With the widespread application of machine learning models across various domains, ensuring that these models are not illegally copied, redistributed, or used without authorization becomes crucial. Model watermarking embeds specific identification information into the model, allowing the original owner to track and prove ownership if the model is misappropriated. Watermark embedding methods include directly modifying model parameters or creating specific trigger datasets that cause the model to exhibit abnormal prediction behavior when processing these data. Watermark verification can be done through white-box (direct access to model parameters) or black-box (simply via the input-output interface of the model) techniques to confirm the watermark's existence [56]. With ongoing research, various watermarking methods have emerged, including parameter-based watermarks, trigger data point-based watermarks, and leveraging the backdoor characteristics of neural networks for watermarking [57].

3) Communication Privacy Protection: Trusted Execution Environment (TEE) [20] is a secure computing environment that provides an isolated execution space to protect code and data from external software and hardware attacks or unauthorized access. To protect sensitive operations and guarantee the security and integrity of code executed and data processed inside TEE, TEE typically makes use of hardware-supported security capabilities. The concept of TEE originated in smartphones and embedded systems to protect sensitive information such as payments and personal data. For instance, ARM TrustZone technology is an early TEE implementation that divides the system into secure and normal worlds using hardware support. As open-source software and hardware continue to advance, the RISC-V architecture has garnered significant attention due to its flexibility and openness. TEE implementations on the RISC-V architecture, such as the Keystone framework [21], provide a customizable TEE solution allowing developers to tailor TEE characteristics and functionalities based on specific requirements.

The remainder of the document is arranged as follows. The fundamentals of synchronous, asynchronous FL, and differential privacy are covered in Section II, which also presents the theory of federated learning and differential privacy. In Section III, we summarize the relevant knowledge of differentially private FL, including the tailoring of gradients and the differential privacy at the user and customer levels. We discuss and make suggestions for future research in Section IV. In section V, we give our conclusions.

II. Federated Learning and Differential Privacy-Related Theories

A. *Federated Learning*

Based on the different update strategies in federated learning, the two types of federated learning that we may distinguish are synchronous and asynchronous. In synchronous federated learning, all participants (or clients) must wait for each other to complete their local computations before sending updates to the central server. The global model is then produced by the central server integrating these updates. This approach ensures that all participants use the same or similar data for training in each round, but it can lead to inefficiencies as it requires waiting for the slowest participant (i.e., the straggler). In contrast, asynchronous federated learning is characterized by its asynchronous update process. The central server can receive and immediately integrate updates from any participant that is ready, without waiting for all participants to complete. This design improves system efficiency and scalability; however, it also introduces new challenges, such as handling data inconsistency and model update delays.

1) Synchronous federated learning optimization algorithm: Data privacy protection is federated learning's primary goal. and security, improve model training efficiency and address the problem of data silos. Stated differently, its goal is to optimize data use across many devices to improve user experience while maintaining the highest level of security and confidentiality for user data. Nowadays, deep learning has made extensive use of optimization based on the stochastic gradient descent (SGD) algorithm. and can also be applied in simple federated learning scenarios. The system architecture diagram of Synchronous federated learning is shown in Fig. 1.

Each client in FedSGD [22] separately computes the loss function's gradient using its dataset, and then transmits that gradient to a main server [23]. Next, the central server combines these gradients (sometimes by averaging them) and updates the global model parameters. All clients receive the revised model parameters back, and they use these new values to continue computing their local gradients. The model is iterated through till it merges.

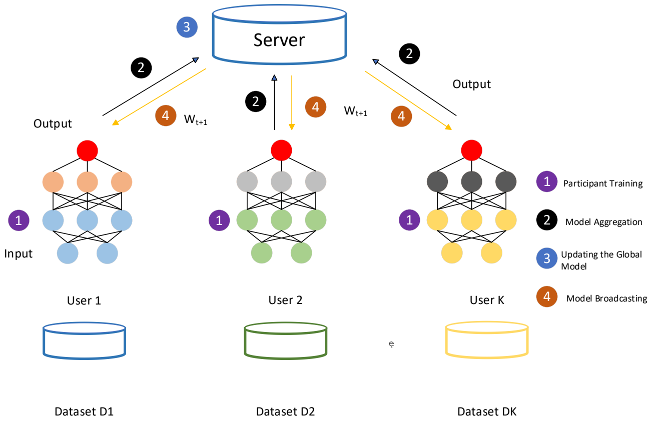

Fig. 1. Schematic diagram of the system architecture of synchronous federated learning.

Building on this, the federated averaging algorithm (FedAvg) was introduced in [24], which combines local stochastic gradient descent computations on clients with model averaging on the server. Local model updates are carried out by clients, and the modified values from every client are averaged by the central server, taking into account the quantity of local updates completed. Each client can independently update its model parameters multiple times before sending the updated parameters to the central server for weighted averaging. The specific formula is represented as follows:

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k = w_t - \eta \nabla f(w_t) \tag{1}$$

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k} \quad \text{where} \quad \forall k, w_{t+1}^{k} \leftrightarrow w_t - \eta g_k \tag{2}$$

In this case, n_k is the number of local datasets for the *k-th* user, and k is the *k-th* user. Early algorithms in federated learning were easy to grasp in notes, but lacked theoretical assurances in practical applications, necessitating extensive experimentation and validation for different environments and sample scenarios [25]. To address non-iid scenarios, where data distributions among clients are uneven, the FedProx algorithm was proposed. Unlike FedAvg, FedProx adds a regularization term to the client-side loss function (while considering the central model) to prevent overfitting during local iterations. Subsequently, [26] introduced a control algorithm for situations where local iteration and edge computing resources are limited in federated learning. By finding the ideal ratio between local updates and global parameter aggregation, this technique maximizes client participation in central aggregation by figuring out how frequently local iterations should occur. Addressing the issue of varying computational capabilities among multiple clients [27], the FedNova algorithm was proposed, assuming heterogeneous client computing resources (i.e., different capacities for local iterations). In study [28], personalized weighting of model parameters per layer on the central server was achieved through a hypernetwork. This strategy entails relearning the model parameters for every client at each layer and minimizing the loss by calculating the disparity between each client's model and the central model from the previous round. Subsequently, to reduce communication costs, layers with significant locally retained weights were excluded from federated participation.

2) Asynchronous federated learning optimization algorithm: Widespread 5G network rollout and quick hardware development are improving the connectivity and computing abilities of heterogeneous devices, such as edge and IoT gadgets and opening up new application areas [29]. Federated learning is gradually integrating functionalities learned from other devices to improve model quality.

However, when federated learning is applied on resource-constrained devices using classical learning methods, several disadvantages become apparent. Due to the presence of heterogeneous devices, the aggregation server needs to wait for updates from different devices, which may unexpectedly go offline due to instability. Faster devices in federated learning training rounds have to wait for slower devices to finish calculations, resulting in low resource utilization due to device performance differences (device heterogeneity) and uneven data distributions (data heterogeneity).

The inefficiency of current node selection algorithms often leads to the involvement of few capable devices. Security and privacy vulnerabilities are also concerns. Security risks like data poisoning and backdoor access might affect traditional federated learning techniques. Privacy concerns also surface because of possible data leaks that occur during training.

Asynchronous federated learning (AFL) offers an answer to these problems. A novel federated learning mechanism called Fed2A was proposed in [30], designed specifically for asynchronous and adaptive modes. Fed2A uses three adaptive methods and a two-phase asynchronous learning approach to support AFL successfully. Specifically, one of the core formulas of Fed2A for global model aggregation is as follows:

$$w_{t+1} = \sum_{l=1}^{L}\sum_{k=1}^{K}(\alpha_{l_k} \times w_{l_k}), \quad \alpha_{l_k} \leftarrow g(t_k, t, w_{l_t}, w_{l_k}) \tag{3}$$

This formula illustrates how Fed2A considers the heterogeneity of time and information during global model aggregation. Here, W_{t+1} represents the global model at global round $t+1$, L is the number of layers in the DNN being trained, K is the total number of participating clients, W_{lk} is the local model parameters of client k at layer l, a_{lk} is the aggregation weight of client k at layer l, g is a function used to compute the aggregation weight. This function considers the generation time t_k of the client's local model, the parameters W_{lk} of the current global model at layer l, and the current global model reception time t.

Regarding the three key challenges of federated learning—edge heterogeneity, non-iid data distribution, and communication resource constraints proposed a mechanism called Grouped Asynchronous Federated Learning (FedGA) [31]. They introduced the Magic-Mirror Method (MMM) scheduling strategy within groups to optimize the completion time of model updates in a single round. By designing scheduling algorithms that determine the order of model uploads and downloads, the system achieves computing-at-the-edge while communicating, enhancing adaptability to heterogeneous edges.

Regarding federated learning (FL) in wireless network scenarios, article [32] proposes an asynchronous FL framework. It addresses the slow startup issue (stragglers) inherent in traditional synchronous FL by implementing periodic aggregation to enhance training efficiency. The article describes the process of global model aggregation as follows:

$$w_{t+1} = w_t + \sum_{k \in \Pi(t)} \frac{|S_k|}{|S|} \Delta w_k(t) \tag{4}$$

The formula indicates that the global model w_{t+1} at global round $t+1$ is obtained by aggregating the current global model w_t with the aggregated local model updates $\Sigma k \in \Pi(t)$. The collection of devices slated to submit model changes at global round t is denoted by $\Pi(t)$ in this instance, $|S_k|$ is the size of device k local training dataset, $|S|$ is the total size of training datasets across all devices, and $\triangle w_k(t)$ denotes the model update completed by device k in its local round t.

To address asynchronous update issues, the article introduces an age-aware aggregation weight design, formulated as follows:

$$\alpha_k(t) = \frac{|S_k| \cdot \delta^{ALU_k(t)}}{\sum_{j \in \Pi(t)} |S_j| \cdot \delta^{ALU_j(t)}} \tag{5}$$

In this formula, $\alpha_k(t)$ represents the aggregation weight of device k at global round t. $ALU_{k(t)}$ denotes the age of device k's local model update, which is the number of iterations since it last received the global model. Δ is a constant used to adjust the influence of age on the weight.

MAPA-S and MAPA-C are two conceptually justified multi-stage adaptive privacy algorithms that were created by the authors of [33] for use in asynchronous federated learning (AFL) scenarios. By utilizing fading clipping thresholds during model convergence to lessen unnecessary noise and enhance learning performance, these algorithms aim to increase the ratio of protecting privacy to model efficacy.

The multi-stage adaptive clipping threshold adjusts the clipping threshold adaptively during training using a decaying clipping threshold θ_c . This approach reduces noise, where θ_c is a decay factor of the initial clipping threshold c. These algorithms enhance model utility while preserving privacy by adjusting clipping thresholds and learning rates. Through adaptive tweaking of these parameters at different training phases, MAPA-S and MAPA-C can more accurately balance privacy protection with functionality.

$$\gamma = \min\left\{\frac{\Delta_{\sigma,K}}{K^2 L \Delta_{\sigma,\tau_C} \tau_C}, \frac{\theta_G^2}{8L\Delta_{\sigma,K}}, \frac{1}{4L}\right\} T \geq \frac{8\Gamma}{\gamma \theta_G^2} \tag{6}$$

The *gamma* (γ) is the learning rate, and T denotes the total number of global iterations. $\Delta_{\sigma,k}$ and $\Delta_{\sigma,\tau c}$ represent parameters related to the model and data. θ_G is the decay ratio of the clipping threshold, and Γ is the upper bound of the loss function. The initial learning rate and the number of iterations in the initial stage are determined using these calculations and are updated in each new stage based on the current model and data conditions.

B. Differential Privacy

The primary goal of differential privacy is to allow the study of overall properties of a dataset without revealing individual information. Put differently, differential privacy entails introducing noise into original datasets or statistical queries. Sacrificing some data accuracy to provide strict privacy protection for user data. This ensures that attackers cannot determine whether specific individual effects are present in the dataset.

1) Centralized differential privacy: A centralized differential privacy paradigm, differential privacy [36] was first introduced by Dwork et al. in 2006 [34]. The article defines that differential privacy requires a trusted central authority, allowing users to send their data directly to the data center without any modifications. The data received from users is stored on a central server. The central authority, however, has little faith in outsiders or data analysts. Therefore, the central authority uses differential privacy to obscure the source dataset before answering statistical inquiries for analysis from outside parties. Centralized differential privacy is the term used to describe this kind of differential privacy implementation.

In general, differential privacy uses strict mathematical definitions to limit this probability gap, as defined in Definition 1:

Definition 1 (ε-differential privacy): A randomized mechanism *M* satisfies ε-differential privacy (ε > 0) if and only if for any adjacent input datasets S and S' and for any possible output value set *R*,the following holds:

$$Pr[M(S)\in R] \le e^{\varepsilon}\cdot Pr[M(S')\in R \tag{7}$$

Definition 2 *(ε, δ)*-differential privacy): A randomized mechanism M satisfies *(ε, δ)*-differential privacy *(ε > 0, δ > 0)* if and only if for any adjacent input datasets *S* and *S'* and for any possible the following holds:output value set R.

$$Pr[M(S)\in R] \le e^{\varepsilon}\cdot Pr[M(S')\in R]+\delta \tag{8}$$

Subsequently, in [35], the authors proposed the Laplace mechanism, a widely recognized differential privacy technique. Through the introduction of random noise into numerical statistical results, this method safeguards individual privacy. A zero-centered Laplace distribution is used to sample the noise. A precise scale parameter selection is necessary for the Laplace distribution in order to guarantee adherence to the stringent requirements of differential privacy. This scale parameter is closely related to the sensitivity of the statistical query, which represents the maximum possible change in query results in the worst-case scenario. As a result, the Laplace mechanism takes the query's sensitivity into account while determining the right amount of noise, enabling the publication of approximate statistical data without disclosing personal information.

Definition 3 (Sensitivity of a Statistical Function): For any numerical statistical function f : D^N→R the sensitivity is as follows:

$$\Delta f := \max_{S,S'\in D^N} |\, f(S) - f(S')\,| \tag{9}$$

2) Local differential privacy: By doing away with the need for a certified server, local differential privacy, or LDP, is a decentralized enhancement over hierarchical approaches. In this approach, a randomized method is used to locally randomize each data item that is disseminated among N user interfaces. The information that has been collected is then safely sent via an encrypted link to the server. The server compiles the information and applies the appropriate adjustment algorithm to produce objective estimations of statistical quantities. The local randomization process at the client side ensures that every data item received by the server is unique, hence the LDP model does not rely on the server being trusted.

Definition 4 (ε -LDP): If and only if the following true for any feasible output value y and any pairings of input values *V* and *V′* , then the randomization method M fulfills ε -LDP (ε > 0):

$$Pr[M(V)=y] \le e^{\varepsilon}\cdot Pr[M(V')=y] \tag{10}$$

Definition 5 (ε , δ)-LDP: A randomized mechanism M satisfies (ε , δ)-LDP (ε > 0, δ > 0) if and only if for any input value pairs V and V′ and for any possible output value y, the following holds:

$$Pr[M(V)=y] \le e^{\varepsilon}\cdot Pr[M(V')=y]+\delta \tag{11}$$

The Harmony system was presented by the authors in [37]. It is a useful, precise, and effective system that is mainly intended for gathering and evaluating data from users of smart devices while meeting LDP requirements. Multidimensional data with both numerical and category qualities might benefit from harmony. In addition to sophisticated machine learning tasks like linear regression, logistic regression, and SVM classification, it provides fundamental statistics like mean and frequency estimates. Additionally, the authors discuss the limitations of existing LDP solutions and propose improvement methods such as mini-batch gradient descent and dimensionality reduction techniques to enhance the performance of machine learning models under LDP constraints. The article concludes by exploring potential applications of Harmony in practical settings and identifying future research directions, including its deployment in real-world scenarios like diagnostic information reporting applications for Samsung smartphones.

In traditional Local Differential Privacy (LDP) techniques, the privacy budget ε is typically allocated to related attributes or processed through sampling methods for high-dimensional data. However, these methods have some limitations. First, allocating the privacy budget evenly to all attributes reduces the density of useful information, thereby affecting the utility of the data. Second, attributes in high-dimensional data often have correlations, and existing models do not fully utilize these correlations to optimize the balance between privacy protection and data utility.

The authors of [38] suggested Univariate Dominance Local Differential Privacy (UDLDP), a novel LDP model, to solve these problems. Through the quantification of attribute correlations, the UDLDP model optimizes the allocation of the privacy budget. Specifically, instead of just spreading the budget uniformly, the UDLDP model permits a more precise distribution of the privacy budget on each associated characteristic via a correlation-bounded perturbation method. This effectively gets around the drawbacks of conventional techniques. To further enhance sampling, a widely used bandwidth reduction method in sensor networks and the Internet of Things, this research extends the correlation-bounded perturbation mechanism. The research further improves the correlation-bounded perturbation mechanism with sampling by finding the optimal sampling probability distribution method with regard to of data utility.

3) Rényi differential privacy: A notion of privacy based on Rényi divergence is called Rényi Differential Privacy (RDP). Rényi divergence is a tool for measuring the difference between two probability distributions and can be viewed as a generalization of Kullback-Leibler divergence. RDP defines a new measure of privacy loss using Rényi divergence, providing a more flexible and fine-grained way to quantify privacy loss.

Definition 6: Given two adjacent datasets D and D′ , which differ by one data point, and a random mechanism M that outputs distributions P and Q, respectively. If for all $\alpha > 1$, the mechanism M satisfies the following inequality, it is said to satisfy (α, ϵ)-Rényi differential privacy:

$$D_{\alpha}(P \| Q) = \frac{1}{\alpha - 1} \log \mathrm{E}_{x \sim Q}\left[\left(\frac{P(x)}{Q(x)}\right)^{\alpha}\right] \leq \epsilon \tag{12}$$

Here, $D_{\alpha}(P||Q)$ represents the α-order Rényi divergence between distributions P and Q. This measure captures privacy loss more precisely by considering higher-order moments of the distributions, providing tighter bounds compared to traditional differential privacy measures.

In order to train deep neural networks to address non-convex optimization problems while ensuring privacy, [39] first presented a revolutionary algorithmic technique. The authors developed an improved Stochastic Gradient Descent (SGD) algorithm that incorporates privacy protection at each step by using gradient clipping and noise addition to control the dependence on individual data points during training. Additionally, the paper introduced a novel privacy loss estimation method called Moments Accountant, which offers tighter privacy guarantees than traditional differential privacy analyses. This method provides a more accurate estimation of the algorithm's privacy cost by tracking higher-order moments of privacy loss.

Based on the method of tracking higher-order moments of privacy loss, Ilya Mironov et al. proposed an extended framework for Differential Privacy (DP) based on Rényi divergence in study [40], known as Rényi Differential Privacy (RDP). RDP aims to improve existing differential privacy techniques by providing a more granular measure of privacy loss. The core of RDP is quantifying slight changes in the output distribution of randomized algorithms. Compared to traditional differential privacy, RDP offers more precise privacy loss estimation by considering higher-order moments of the distribution. This measure allows for more detailed analysis of the algorithm's output while protecting privacy.

Additionally, in study [41], the authors focused on studying the Sampling Gaussian Mechanism (SGM), a widely used technique in machine learning that combines data subsampling and Gaussian noise addition to provide privacy protection. They proposed a numerically stable procedure to accurately compute the RDP of SGM. The researchers demonstrated that SGM satisfies (α,ϵ)-RDP under specific conditions and provided an almost tight closed-form bound. This work fills previous research gaps and unifies the understanding of SGM's privacy properties. The authors provided deep insights into understanding and applying RDP, especially in analyzing and designing privacy-preserving machine learning algorithms. By accurately computing the RDP of SGM, this research advances theoretical development and offers practical tools and guidance for privacy protection in real-world applications.

In "Hypothesis Testing Interpretations and Rényi Differential Privacy," the authors proposed a new perspective by interpreting differential privacy through statistical hypothesis testing. Within this framework, differential privacy ensures that no test can simultaneously have high significance (low Type I error rate) and high power (low Type II error rate). Additionally, the authors provided improved conversion rules from RDP to (ϵ,δ)-DP and explored the relationship with Gaussian Differential Privacy (GDP). Finally, they proposed a sufficient and necessary condition to ensure that a quasi-convex divergence is k-generated. By requiring divergences to be defined using a 2-generated function F, this aids in the construction of divergences that support the interpretation of the hypothesis test.

III. Differential Privacy for Federated Learning

A. Federated Learning with Differential Privacy with Different Gradient Clipping

Several participants can train models on their local data using the central server in traditional Federated Learning (FL), eliminating the requirement to centralize the data on a single server. However, during the transmission of model parameters, if communication is not encrypted or if there are vulnerabilities, it may be susceptible to eavesdropping or tampering. An untrusted central server could infer sensitive information by analyzing model updates and gradient information. Differential Privacy (DP) effectively addresses these issues by adding noise to gradients to prevent such information leaks. However, in study [42], it was first proposed to use a fixed gradient clipping approach.

On one hand, the amount of noise added in fixed gradient clipping differential privacy remains constant throughout the training process. This could lead to excessive negative impacts on model performance due to noise in the later stages of model training, thus affecting the model's usability. On the other hand, a fixed clipping threshold may not be suitable for all datasets or training scenarios. Different data distributions and models may require different clipping strategies to achieve optimal privacy protection.

1) Federated learning with fixed gradient clipping differential privacy: To address the shortcomings of fixed clipping, the authors in study [43] attempted to solve the issues of parameter privacy protection and high communication costs by combining distinguished differential privacy with gradient trimming in two stages. The trained model's gradients are pruned in the first stage of the proposed IsmDP-FL, and the key variables that are chosen are then given differential privacy. To finish the federated learning manage, gradient trimming is carried out in the subsequent phase while the data is being sent to the server for consolidation. The final result is then sent back to the client. Comparing the IsmDP-FL algorithm to other approaches, experimental results showed that it achieves higher model accuracy while maintaining high communication efficiency and model privacy.

In study [44], the authors proposed a layer pruning method based on gradient correlation to further reduce communication overhead. Instead of uniformly clipping the parameters of all layers, the CLFLDP model uses a layer selection method based on model correlation metrics to choose layers with higher correlation to the global model for upload, excluding those with lower correlation. By using a Top-k gradient reduction strategy, the model further decreases the total amount of parameters uploaded inside the chosen layers; only the parameters with the highest gradient values are chosen and uploaded to the server.

2) Federated learning with adaptive gradient clipping differential privacy: In study [45], the AdaCliP algorithm is introduced, with its core innovation being an adaptive clipping mechanism that dynamically adjusts the clipping threshold based on the gradient characteristics of each coordinate. This approach not only reduces unnecessary noise addition but also enhances the model's sensitivity to data during training, thereby improving model accuracy without sacrificing privacy. The implementation of AdaCliP is based on precise control of gradients during the stochastic gradient descent (SGD) process. By introducing dynamic estimates of the mean and standard deviation, the algorithm can adaptively adjust gradient clipping and noise addition at each iteration. Moreover, the convergence analysis provided in the paper offers a solid theoretical foundation for the algorithm's performance.

The authors of study [46] suggest an adaptive clipping technique that modifies the clipping threshold to roughly represent a particular quantile in the updates' norm distribution. This adaptive clipping is implemented using an online gradient descent algorithm by designing a loss function $\ell\gamma(C;X)$ for a random variable X and quantile γ to estimate and update the clipping threshold C. The form of the loss function ensures that the expected value of its derivative reflects the relationship between C and the quantile of X, allowing C to approach the true quantile of X via gradient descent. This approach not only closely tracks the quantile of update norms but is also compatible with techniques like compression and secure aggregation in federated learning, all while consuming minimal privacy budget.

In study [47], the authors propose a novel adaptive differential privacy method that shifts focus away from gradients to determine the amount of noise injected based on the importance of features. Less noise is injected for important features, whereas more noise is added for less important ones. The paper introduces two adaptive methods: Sensitivity-Based Method: This method evaluates the importance of features by computing changes in model accuracy after adding noise. After updating local parameters, the client computes and stores the model accuracy as a reference. The weights associated with each input characteristic are then increased by noise and the accuracy of the new model is computed. Feature importance is determined by comparing the accuracy before and after noise addition. Variance-Based Method: This approach assumes that weights associated with more important features undergo greater changes during training. A value that is equal to the influence on output is generated by computing the variance of the weights attached to each input neuron. Following the determination of the significance of each feature, the differential privacy parameters are tuned to balance privacy protection and model performance by adding more noise to less significant characteristics and less noise to key ones.

B. User-level and Sample-level Differential Privacy Federated Learning

1) User-level differential privacy federated learning: A privacy-preserving method used in federated learning to safeguard participants' privacy inside the framework is called user-level differential privacy. In this configuration, several users work together to train a machine-learning model while maintaining the privacy of their personal information. In user-level differential privacy for federated learning, all user data is usually protected, which means that all sample gathering on a user's device is protected [48]. This is significant because, even if an attacker manages to access the data of every other user, they will still be unable to deduce each individual user's data from the combined findings.

To protect all data of each user, user-level differential privacy in federated learning requires adding noise to the model updates computed locally by each user, in order to satisfy user-level differential privacy requirements. This means that after local training, noise is added to the gradients or model parameter updates of the entire dataset.

In study [49], Mcmahan et al. first proposed DP-FedAvg and DP-FedSGD, where sampling is performed on the client side, and noise addition occurs centrally. Sensitivity computation is based on the sampling rate and the federated weights of each client. In the same period, another article [50] distinguished itself from DP-FedAvg by having client-side model uploads trimmed at the central server. The algorithm in this paper achieves client-level differential privacy protection through the aggregation of distorted updates using random sub-sampling and Gaussian mechanisms. The algorithm's secret is to strike a balance between model performance and privacy protection. According to experimental findings, CDPFL can provide client-level differential privacy with a minimum loss in model performance provided there are enough clients involved.

In the paper [51], the authors focus on the scenario where model parameters in federated learning may be analyzed by malicious servers. They propose a User-Level Differential Privacy (UDP) algorithm aimed at enhancing privacy protection in FL. The primary goal of the UDP technique is to obfuscate the relationship between model parameters and users' original data by introducing fake noise to the shared model prior to uploading it to the server. By adjusting the variance of the noise, the algorithm can provide different levels of privacy protection for each mobile terminal, meeting the (ϵ_i, δ_i)-LDP privacy protection standard. Here, ϵ_i and δ_i are privacy parameters associated with the i-th mobile terminal, and by adjusting the variance σi^2 of the noise, the level of privacy protection can be controlled. According to the analysis in the paper, the specific formula for computing the noise variance is as follows:

$$\sigma_i = \frac{\Delta\ell}{\sqrt{2qT\ln(1/\delta_i)}} \cdot \frac{1}{\grave{o}_i} \tag{13}$$

Here, $\Delta\ell$ represents the sensitivity of the local training process, q is the random sampling rate, T denotes the quantity of communication cycles, ε_i denotes the privacy protection parameter, and δi stands for the failure probability.

User-level differential privacy in federated learning ensures privacy protection at the user level in FL. User-level DP focuses on protecting all data of each user or agent, rather than individual data instances. For example, in scenarios such as banks jointly training fraud detection models, user-level DP can protect individual records from any bank from being identified. In scenarios like learning facial recognition models on smartphone apps, user-level DP can protect the privacy of each user as a unit. However, existing user-level DP methods, such as DP-FedAvg based on Gaussian mechanism, often sacrifice model utility because they require trimming of model updates for each agent before uploading, and adding Gaussian noise proportional to the trimming threshold. This can lead to decreased model performance.

To address these issues, in [52], the authors analyzed the reasons behind the significant decrease in model accuracy when ensuring user-level DP using existing methods. They proposed two techniques: Two methods are Local Update Sparsification (LUS) and Local Update Regularization (BLUR). Through the addition of regularization terms to the agents' local objective function, BLUR constrains the L2 standard for local changes. While LUS further reduces the norm of updates by zeroing out values that have minimal impact on local model performance before trimming. Both techniques aim to enhance model utility without compromising privacy.

2) Sample-level differential privacy federated learning: Differential privacy at the sample level Federated learning is a machine learning paradigm that safeguards the privacy of individual data while enabling several users to work together on model training. The core of this paradigm ensures that each participant's data remains private even in distributed data environments, preventing data leakage to other participants or potential attackers.

In traditional federated learning, although data does not need to be centrally stored or processed, there remains a risk of privacy leakage. Attackers could potentially infer information about individual clients by analyzing shared model updates or gradient information among clients. To address this issue, researchers have proposed sample-level differential privacy federated learning, aiming to provide privacy protection for each data record of every client.

The research in [53] further advances research in this field. The authors introduce the concept of federated ε-differential privacy, a novel privacy protection measure based on the Gaussian differential privacy framework. It focuses on the record level, protecting each client's unique data record from other clients' attacks by offering privacy protection. The PriFedSync framework proposed in the paper is a generic private federated learning framework capable of accommodating various existing federated learning algorithms and demonstrating its effectiveness in achieving federated ε-differential privacy. The paper also conducts experiments in computer vision tasks, demonstrating that while ensuring privacy, the model can still maintain high predictive performance. This indicates the potential of sample-level differential privacy federated learning in practical applications, especially in fields such as healthcare and finance where data privacy requirements are stringent.

In study [54], the authors propose a novel sample-level differential privacy federated learning method—DP-SCAFFOLD, aiming to address both data heterogeneity and privacy protection issues. This method integrates differential privacy constraints into the popular SCAFFOLD algorithm to achieve sample-level privacy protection for participating users. In scenarios without trusted intermediaries, users communicate with "honest but curious" servers. This approach not only targets privacy protection from third-party observations of the final model but also ensures that "honest but curious" servers themselves cannot accurately reconstruct user data in the absence of a trusted intermediary. The paper provides in-depth analysis of the convergence of the DP-SCAFFOLD algorithm, demonstrating its convergence under convex and non-convex objectives. Additionally, using Rényi differential privacy (RDP) tools, the authors formally describe the privacy-utility trade-offs of DP-FedAvg and DP-SCAFFOLD algorithms at different privacy protection levels. Results show that DP-SCAFFOLD exhibits superiority over DP-FedAvg especially in scenarios with a large number of local updates or high data heterogeneity.

In study [55], the authors address the model evaluation issue in federated learning (FL) by proposing a novel algorithm to compute the AUC metric while ensuring the privacy of labels. AUC is a critical metric for assessing the performance of classification models, and its computation process can potentially expose sensitive information within the dataset. To mitigate this issue, the algorithm in the paper employs differential privacy techniques, particularly the Laplace mechanism, to inject appropriate noise into intermediate results during the computation process.

$$\Pr[M(D) \in S] \le e^{\grave{o}} \cdot \Pr[M(D') \in S] + \delta \tag{14}$$

Here, $\mathcal{M}$ represents the random mechanism, $\mathcal{D}$ and $\mathcal{D}'$ are two adjacent datasets differing in a single sample's label. S is a subset of the output results, ϵ is the privacy budget used to quantify the strength of privacy protection, and δ is a small non-negative value used for handling boundary cases. Specifically, the definition of label differential privacy proposed in the paper emphasizes sensitivity to changes in individual sample labels.

In the setting of federated learning (FL), each client independently predicts and computes statistics on their data, then sends the noisy statistics to the server. Without directly accessing the original labels, the server aggregates these noisy statistics to compute the AUC. This method guarantees the correctness of the model evaluation while simultaneously safeguarding the confidentiality of customer data.

IV. Discussion and Recommendations for Future Research

In this section, we will first discuss some key issues, and then introduce our recommendations for future research.

We first discuss the following key issues: Differential Privacy with Federated Learning (DPFL) is moving towards a more efficient and personalized direction, which helps to achieve a better balance between protecting privacy and maintaining model performance. We believe that future research should continue to explore more advanced adaptive privacy protection mechanisms. At the same time, we find that user-level and sample-level differential privacy each have their advantages. Researchers should choose the appropriate type based on specific application scenarios and privacy needs and can explore a hybrid privacy protection strategy that combines the two. Asynchronous DPFL has potential in dealing with the heterogeneity of devices in practical scenarios, but still needs to address the challenges of model convergence and privacy protection, which provides an important direction for our future research. We emphasize that developing personalized privacy protection strategies is crucial for improving the practicality of DPFL, and future research should focus on how to meet the differentiated privacy needs of individuals while protecting overall privacy. Finally, different application scenarios have different needs for privacy and utility, and future research should further explore the best balance point for different application scenarios.

Differential privacy federated learning combines the advantages of data privacy protection and distributed machine learning, making it a current hotspot in research. Most current research focuses on synchronous updating federated learning frameworks, but in practical applications, the computing and communication resources among participants are asynchronous, posing numerous unresolved challenges in this field. Existing federated learning frameworks often assume synchronous model updates across all participants, which is impractical in real-world scenarios. Current privacy protection strategies are typically one-size-fits-all and fail to fully consider personalized privacy needs among different participants. Validation of differential privacy federated learning in real-world applications and its cross-domain applications remain relatively limited.

In order to support the asynchronous computing and communication resources among participants, future research should concentrate on developing effective asynchronous communication protocols. This approach ensures model convergence and performance while maximizing the utilization of each participant's computing resources. Furthermore, future studies can explore asynchronous federated learning differential privacy and personalized federated learning to further advance this field. While current federated learning frameworks assume synchronous updates, the reality of varying computing and communication resources among participants necessitates efficient asynchronous communication protocols. Customized privacy protection strategies can also be explored to cater to the different privacy needs and sensitivities among participants, thereby enhancing the flexibility and adaptability of federated learning.

Moreover, applying differential privacy federated learning to more practical domains such as healthcare, finance, and the Internet of Things (IoT) will validate its effectiveness and potential in different application scenarios. By identifying and addressing new challenges through practical applications, continuous improvement and maturation of the technology can be achieved.

These future studies will help overcome the limitations of current research, enhancing the effectiveness and adaptability of differential privacy federated learning in practical applications. Research on asynchronous federated learning differential privacy will make model training more efficient, personalized privacy protection strategies will meet the specific needs of different participants, and cross-domain applications and validations will drive the application and development of the technology in more practical scenarios. These studies will provide new perspectives for theoretical development and offer a more solid foundation for practical applications

V. Conclusion

Federated Learning (FL) as an innovative distributed machine learning technique has shown enormous potential in protecting data privacy and security. However, FL still faces numerous privacy and security challenges in practical applications. This paper provides a detailed review of Differential Privacy Federated Learning (DPFL). After outlining the basic concepts of differential privacy and federated learning, we categorize their integration. Subsequently, we discuss DPFL using different gradient clipping strategies, including fixed clipping and adaptive clipping methods, to enhance the protection capability and efficiency of differential privacy. Additionally, we explore the differences between user-level and sample-level differential privacy in federated learning. This paper aims to assist researchers in identifying and developing optimal algorithms for DPFL, while also pointing out future research directions. These include designing asynchronous communication protocols, exploring personalized privacy protection strategies, and expanding the application of DPFL to broader practical scenarios. Through these studies, we hope to overcome the limitations of current research, enhance the effectiveness and adaptability of DPFL in practical applications, and provide a solid theoretical and practical foundation for efficient distributed learning while preserving user privacy.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China (No. 62302540), with author F.F.S. For more information, please visit their website at https://www.nsfc.gov.cn/ . Additionally, it is also funded by the Open Foundation of Henan Key Laboratory of Cyberspace Situation Awareness (No. HNTS2022020), where F.F.S is an author. Further details can be found at http://xt.hnkjt.gov.cn/data/pingtai/. The research is also supported by the Natural Science Foundation ofHenan Province Youth Science Fund Project (No. 232300420422), and for more information, you can visit https://kjt.henan.gov.cn/2022/09-02/2599082.html.

REFERENCES

[1] Rauniyar, A., Hagos, D. H., Jha, D., Håkegård, J. E., Bagci, U., Rawat, D. B., & Vlassov, V. (2023). Federated learning for medical applications: A taxonomy, current trends, challenges, and future research directions. IEEE Internet of Things Journal.].

[2] CHATTERJEE, Pushpita; DAS, Debashis; RAWAT, Danda B. Federated Learning Empowered Recommendation Model for Financial Consumer Services. IEEE Transactions on Consumer Electronics, 2023.

[3] Chen, J., Xue, J., Wang, Y., Huang, L., Baker, T., & Zhou, Z. (2023). Privacy-Preserving and Traceable Federated Learning for data sharing in industrial IoT applications. Expert Systems with Applications, 213, 119036.

[4] Yang, R., Ma, J., Zhang, J., Kumari, S., Kumar, S., & Rodrigues, J. J. (2023). Practical feature inference attack in vertical federated learning during prediction in artificial internet of things. IEEE Internet of Things Journal.

[5] Gupta, P., Yadav, K., Gupta, B. B., Alazab, M., & Gadekallu, T. R. (2023). A novel data poisoning attack in federated learning based on inverted loss function. Computers & Security, 130, 103270.

[6] Wei, HUO.,Yu, YU., Kang, YANG., Zhongxiang ZHENG., Xiangxue LI., & Li, YAO.(2023).Privacy-preserving cryptographic algorithms and protocols: a survey on designs and applications.Scientia Sinica(Informationis)(09),1688-1733.

[7] HAN, Wei-Li. , SONG Lu-Shan.,RUAN, Wen-Qiang;LIN, Guo-Peng.,WANG, Zhe-Xuan.(2023).Secure Multi-Party Learning:From Secure Computation to Secure Learning.Chinese Journal of Computers(07),1494-1512.

[8] HANG, J., CHEN, J.,WU, W.,&FENG,Y.Privacy-Preserving Principal Component Analysis Based on Homomorphic Encryption.Computer Science1-13.

[9] Bai,L., ZHU, Y., LI, Y., WANG, S., & Yang Xiao-Qi. Progress in the research of total homomorphic encryption. Computer Research and Development 1-19.

[10] Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23 (pp. 409-437). Springer International Publishing.

[11] Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., ... & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. IEEE transactions on information forensics and security, 15, 3454-3469.

[12] Alabdulatif, A., Kumarage, H., Khalil, I., & Yi, X. (2017). Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption. Journal of Computer and System Sciences, 90, 28-45.

[13] Chen, Y., Lu, W., Qin, X., Wang, J., & Xie, X. (2023). Metafed: Federated learning among federations with cyclic knowledge distillation for personalized healthcare. IEEE Transactions on Neural Networks and Learning Systems.

[14] Wu, C., Wu, F., Lyu, L., Huang, Y., & Xie, X. (2022). Communication-efficient federated learning via knowledge distillation. Nature communications, 13(1), 2032.

[15] Wen, H., Wu, Y., Hu, J., Wang, Z., Duan, H., & Min, G. (2023). Communication-efficient federated learning on non-IID data using two-step knowledge distillation. IEEE Internet of Things Journal.

[16] Gong, X., Sharma, A., Karanam, S., Wu, Z., Chen, T., Doermann, D., & Innanje, A. (2021). Ensemble attention distillation for privacy-preserving federated learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 15076-15086).

[17] Wu, C., Wu, F., Lyu, L., Huang, Y., & Xie, X. (2022). Communication-efficient federated learning via knowledge distillation. Nature communications, 13(1), 2032.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.

[19] J. Ortigosa-Hernández, I. Inza, and J. A. Lozano, “Measuring the class-imbalance extent of multi-class problems,” Pattern Recognit. Lett., vol. 98, pp. 32–38, Oct. 2017.

[20] Mo, F., Haddadi, H., Katevas, K., Marin, E., Perino, D., & Kourtellis, N. (2021, June). PPFL: privacy-preserving federated learning with trusted execution environments. In Proceedings of the 19th annual international conference on mobile systems, applications, and services (pp. 94-108).

[21] Lee, D., Kohlbrenner, D., Shinde, S., Asanović, K., & Song, D. (2020, April). Keystone: An open framework for architecting trusted execution environments. In Proceedings of the Fifteenth European Conference on Computer Systems (pp. 1-16).

[22] McMahan, H. B., Moore, E., Ramage, D., & y Arcas, B. A. (2016). Federated learning of deep networks using model averaging. arxiv preprint arxiv:1602.05629, 2, 2.

[23] DUAN, X.,CHEN, G.,CHEN, A.,CHEN, C., & JI, W.,College of Information and Navigation, Air Force Engineering University;(2024).Review of Research on Information Security in Federated Learning.Computer Engineering and Applications(03),61-77.

[24] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[25] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2, 429-450.

[26] Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., & Chan, K. (2019). Adaptive federated learning in resource constrained edge computing systems. IEEE journal on selected areas in communications, 37(6), 1205-1221.

[27] Wang, J., Liu, Q., Liang, H., Joshi, G., & Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. Advances in neural information processing systems, 33, 7611-7623.

[28] Ma, X., Zhang, J., Guo, S., & Xu, W. (2022). Layer-wised model aggregation for personalized federated learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10092-10101).

[29] Xu, C., Qu, Y., Xiang, Y., & Gao, L. (2023). Asynchronous federated learning on heterogeneous devices: A survey. Computer Science Review, 50, 100595.

[30] [Fed2A_Federated_Learning_Mechanism_in_Asynchronous

[31] MA, Q., JIA, Q,LIU, J.,XU, H.,XIE, R., & HUANG, Tao.,(2023).Client grouping and time-sharing scheduling for asynchronous federated learning in heterogeneous edge computing environment.Journal on Communications(11),79-93.

[32] Hu, C. H., Chen, Z., & Larsson, E. G. (2023). Scheduling and aggregation design for asynchronous federated learning over wireless networks. IEEE Journal on Selected Areas in Communications, 41(4), 874-886.

[33] Li, Y., Yang, S., Ren, X., Shi, L., & Zhao, C. (2023). Multi-stage Asynchronous Federated Learning with Adaptive Differential Privacy. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[34] Dwork, C. (2006, July). Differential privacy. In International colloquium on automata, languages, and programming (pp. 1-12). Berlin, Heidelberg: Springer Berlin Heidelberg.

[35] Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 9(3–4), 211-407.

[36] McSherry, Frank, and Kunal Talwar. "Mechanism design via differential privacy." in 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). IEEE, 2007.pp.94-103.

[37] Nguyên, T. T., Xiao, X., Yang, Y., Hui, S. C., Shin, H., & Shin, J. (2016). Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053.

[38] Du, R., Ye, Q., Fu, Y., & Hu, H. (2021, July). Collecting high-dimensional and correlation-constrained data with local differential privacy. In 2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) (pp. 1-9). IEEE.

[39] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318).

[40] Mironov, I. (2017, August). Rényi differential privacy. In 2017 IEEE 30th computer security foundations symposium (CSF) (pp. 263-275). IEEE.

[41] Mironov, I., Talwar, K., & Zhang, L. (2019). R\'enyi differential privacy of the sampled gaussian mechanism. arXiv preprint arXiv:1908.10530.

[42] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.

[43] Li, Y., Du, W., Han, L., Zhang, Z., & Liu, T. (2023). A Communication-Efficient, Privacy-Preserving Federated Learning Algorithm Based on Two-Stage Gradient Pruning and Differentiated Differential Privacy. Sensors, 23(23), 9305.

[44] Chen, S., Yang, J., Wang, G., Wang, Z., Yin, H., & Feng, Y. (2024). CLFLDP: Communication-efficient layer clipping federated learning with local differential privacy. Journal of Systems Architecture, 148, 103067.

[45] Pichapati, V., Suresh, A. T., Yu, F. X., Reddi, S. J., & Kumar, S. (2019). Adaclip: Adaptive clipping for private sgd. arXiv preprint arXiv:1908.07643.

[46] Andrew, G., Thakkar, O., McMahan, B., & Ramaswamy, S. (2021). Differentially private learning with adaptive clipping. Advances in Neural Information Processing Systems, 34, 17455-17466.

[47] Talaei, M., & Izadi, I. (2024). Adaptive Differential Privacy in Federated Learning: A Priority-Based Approach. arXiv preprint arXiv:2401.02453.

[48] Shi, Y., Liu, Y., Wei, K., Shen, L., Wang, X., & Tao, D. (2023). Make landscape flatter in differentially private federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 24552-24562).

[49] McMahan, H. B., Ramage, D., Talwar, K., & Zhang, L. (2017). Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963.

[50] Geyer, R. C., Klein, T., & Nabi, M. (2017). Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557.

[51] Wei, K., Li, J., Ding, M., Ma, C., Su, H., Zhang, B., & Poor, H. V. (2021). User-level privacy-preserving federated learning: Analysis and performance optimization. IEEE Transactions on Mobile Computing, 21(9), 3388-3401.

[52] Cheng, A., Wang, P., Zhang, X. S., & Cheng, J. (2022). Differentially private federated learning with local regularization and sparsification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10122-10131).

[53] Zheng, Q., Chen, S., Long, Q., & Su, W. (2021, March). Federated f-differential privacy. In International conference on artificial intelligence and statistics (pp. 2251-2259). PMLR.

[54] Noble, M., Bellet, A., & Dieuleveut, A. (2022, May). Differentially private federated learning on heterogeneous data. In International Conference on Artificial Intelligence and Statistics (pp. 10110-10145). PMLR.

[55] Sun, J., Yang, X., Yao, Y., Xie, J., Wu, D., & Wang, C. (2023, June). Dpauc: Differentially private auc computation in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 12, pp. 15170-15178).

[56] Boenisch, F. (2021). A systematic review on model watermarking for neural networks. Frontiers in big Data, 4, 729663.

[57] Zhang, J., Chen, D., Liao, J., Zhang, W., Feng, H., Hua, G., & Yu, N. (2021). Deep model intellectual property protection via deep watermarking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(8), 4005-4020.