WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 5, Number 1, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v5n1.06
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Federated Learning Based on Dimension Selection

Mengyuan Cheng *

School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing, China

ABSTRACT

Federated learning, as a distributed machine learning framework, enables users to collaboratively train models by sharing model parameters without exposing their raw data. However, model parameters may contain privacy-sensitive information, and directly sharing them still poses a risk of user privacy leakage. Local Differential Privacy (LDP) effectively defends against adversaries with arbitrary background knowledge, providing more comprehensive privacy protection. However, the high-dimensional nature of parameters in federated learning presents challenges for the application of LDP. To address this issue, this paper proposes a federated learning algorithm, FDL, that satisfies local differential privacy. The algorithm employs a dimension selection strategy to identify parameter dimensions critical for global aggregation and applies the Laplace mechanism to perturb these dimensions. Compared to traditional methods, the FDL algorithm significantly reduces the number of parameters to be processed and the amount of noise introduced. Theoretical analysis proves that the FDL algorithm satisfies-local differential privacy, and experimental results demonstrate its high usability while ensuring strong privacy protection.

KEYWORDS

Federated Learning; Local Differential Privacy; Dimension Selection

1. INTRODUCTION

With the rapid development of the digital era, machine learning technologies have demonstrated broad application potential in various fields, such as image recognition [1] and natural language processing [2]. Traditional machine learning requires large amounts of data to train models. However, institutions holding data often face limitations in data volume, and due to the sensitive nature of data, data owners are unwilling to share their data with any centralized data collectors, resulting in the significant problem of "data silos." To address this issue, Google introduced the concept of federated learning in 2016.

Federated Learning (FL) is an innovative distributed machine learning framework that allows participants to send only trained model parameters to a central server for aggregation and updates without sharing their raw data [3]. This framework has become a research hotspot in recent years. Although users in FL update their models locally, the shared updated parameters often contain private information, exposing users to the risk of privacy leakage [4, 5]. Currently, privacy-preserving federated learning methods primarily fall into two categories: data encryption [6, 7] and data perturbation [8]. Compared to encryption-based methods, perturbation-based methods are computationally efficient, easy to implement, and suitable for routine deployment in federated learning applications.

As a popular perturbation technique, Differential Privacy (DP) [9] is a robust privacy-preserving model that defends against adversaries with arbitrary background knowledge and provides rigorous, quantifiable privacy guarantees. Local Differential Privacy (LDP), an extension of centralized

differential privacy, introduces the concept of "local data perturbation," where the privatization process is shifted to individual users. This allows users to independently process and protect their sensitive information, achieving more thorough privacy protection [10]. This feature makes LDP well-suited for federated learning.

Many studies have conducted in-depth research on federated learning under local differential privacy. Zhao et al. [11] proposed a multidimensional LDP data collection mechanism to perturb local parameters. However, the noise introduced by this method is proportional to the number of model dimensions. Since machine learning models often have thousands or even millions of parameters [12], this approach leads to an exponential growth in the privacy budget, making it difficult to achieve effective privacy protection. To address the challenges posed by this "curse of dimensionality," studies [14-16] have employed strategies such as random sampling, dimension dropping, and dimension selection to reduce the impact of high-dimensional parameters. However, these methods often significantly affect model accuracy.

To address the aforementioned issues, this paper proposes an efficient federated learning method, FDL, based on local differential privacy. The method introduces a dimension selection mechanism based on the exponential mechanism to tackle the challenges posed by the curse of dimensionality while reducing communication overhead. The main contributions of this work are as follows:

A dimension selection mechanism is introduced, which selects a subset of dimensions for sharing based on the importance of parameters after model training. This approach reduces the number of transmitted parameters, alleviates communication overhead, and mitigates the expansion of privacy budgets caused by increasing dimensions.

The privacy guarantees of the FDL method are theoretically analyzed, and the method is evaluated on real-world datasets. Experimental results demonstrate that the proposed approach achieves high performance under the same privacy protection conditions.

2. RELATED WORK

Federated learning (FL), as a distributed machine learning framework, provides effective solutions to current challenges in artificial intelligence, particularly in addressing data privacy issues in machine learning. While FL can protect privacy to some extent, recent studies [17, 18] have shown that certain attack methods can still disrupt the learning process of FL or steal participants' personal information. Intuitively, cryptographic techniques, such as secure multi-party computation and homomorphic encryption, are often the first choice for addressing privacy concerns. However, these encryption-based approaches inevitably incur high communication and computational costs.

In addition, with the growing interest in differential privacy (DP), an increasing number of studies have focused on applying DP to federated learning algorithms. Shokri and Shmatikov [19] designed and implemented a practical neural network model that allows multiple participants to collaboratively train by performing stochastic gradient descent locally, uploading selected parameters, and applying DP to ensure parameter privacy. However, their experiments used a relatively large privacy budget, resulting in insufficient privacy protection. Abadi et al. [20] also applied DP to safeguard private information, proposing a privacy accounting algorithm to track privacy loss during model training. Geyer et al. [21] developed a differential privacy technique with dynamically adjustable privacy parameters. By introducing Gaussian noise to the server-updated parameters, this method effectively obscured the contributions of individual client datasets. Additionally, it reduced communication costs by uploading only modified parameters.

The aforementioned techniques rely on a trusted third party for implementing DP, but in practice, servers are often untrustworthy. Therefore, Wang et al. [15] introduced local differential privacy (LDP) into federated learning, adopting a multidimensional LDP mechanism to perturb local parameters. While effective, the noise introduced scales with the parameter dimensions. Subsequently,

Wang et al. [14] developed a novel prior random response technique, adding random noise to local updates to obscure user contributions and using random sampling to reduce communication costs. Meanwhile, Shin et al. [22] attempted to reduce parameter dimensions through random projection, but this approach randomly discards some parameter dimensions.

To overcome these limitations, Miao et al. [23] employed compressed sensing techniques to reduce parameter dimensions, albeit at the cost of increased computational overhead. Furthermore, Cui et al. [24] and Wang et al. [25] analyzed parameter or gradient magnitudes to identify and upload critical parameters, reducing dimensions but potentially losing information due to incomplete assessments. Liu et al. [26] proposed an LDP-FL two-stage framework based on stochastic gradient descent, selectively perturbing and uploading the top-k dimensions. However, selecting dimensions solely based on absolute values risks privacy leakage.

3. PRELIMINARY KNOWLEDGE

3.1. Local Differential Privacy (LDP)

DP (Differential Privacy) implements the assumption of maximizing the background knowledge of adversaries. It is a privacy protection technique that provides a strict definition of privacy protection and quantitative evaluation methods. Its implementation relies on a trusted third party. LDP (Local Differential Privacy) is a distributed variant of DP, which guarantees privacy for each local participant and eliminates the assumption of a trusted third party.

Definition 1 (ε-LDP): A random algorithm M satisfies ε-LDP if, for any pair of inputs x and within the domain of the algorithm, and for any possible output $y \in Range(M)$, the following inequality holds:

$$Pr[M(x) = y] <= e^{\varepsilon} Pr[M(x') = y] \tag{1}$$

Where ε is used to quantify the level of privacy protection. The smaller the value of ϵ\epsilonϵ, the stronger the privacy protection provided.

Definition 2 (Exponential Mechanism): Given a utility function u and an output $y \in Range(M)$, for any two possible inputs x and x', the sensitivity of u is defined as $\Delta u = max_{y \in Range(M)} |u(x, y) - u(x', y)|$. The exponential mechanism chooses and outputs ooo based on the following probability, to satisfy ε-LDP:

$$\Pr[M_{EM}(x) = y] = \frac{exp(\frac{\varepsilon u(x, y)}{2\Delta u})}{\sum_{y' \in Range(M)} exp(\frac{\varepsilon u(x, y')}{2\Delta u})} \tag{2}$$

Where the utility function u maps the input to a utility score, the higher the utility score, the higher the probability of the output.

Definition 3 (Sensitivity): For any two adjacent datasets D and D', let Q be a query function. The global sensitivity is defined as:

$$\Delta Q = \max_{D, D'} \left\| Q(D) - Q(D') \right\|_2 \tag{3}$$

Where D and D' represent two datasets that differ by at most one element.

Definition 4 (Laplace Mechanism): Given a dataset D and a query function Q, the Laplace mechanism that satisfies ε −LDP perturbs the query result in the following way:

$$Q_L(D,\varepsilon)=Q(D)+N(0,\frac{\Delta Q}{\varepsilon}) \tag{4}$$

Where $N(0,\frac{\Delta Q}{\varepsilon})$ is the Laplace noise distribution centered at 0 with scale parameter $\frac{\Delta Q}{\varepsilon}$, and Q_L is the query result returned.

Theorem 1 (Sequential Composition) Let there be H perturbation algorithms $M_1(x)$, $M_2(x)$, ..., $M_H(x)$ for the same dataset D, where each algorithm is a differential privacy algorithm with privacy budgets ε_1, ε_2, ..., ε_H respectively. The composed algorithm $M_{1,2,...,H}$ =($M_1(x)$, $M_2(x)$, ..., $M_H(x)$) satisfies the following: $\sum_{i=1}^{H}\varepsilon_i - LDP$.

3.2. Federated Learning

In traditional centralized machine learning frameworks, all data collection and processing must be concentrated in one place. This not only raises risks for data privacy protection but also leads to significant communication overhead due to data transmission. The introduction of federated learning aims to address these issues by enabling local model training at the data source. Instead of aggregating raw data, the model updates are shared and combined, thus achieving both data privacy protection and effective utilization of data resources. Depending on the distribution patterns of sample space and feature space, federated learning can be classified into three main types: horizontal federated learning, vertical federated learning, and federated transfer learning.

This paper primarily focuses on horizontal federated learning, and the flowchart is shown in Figure The specific steps are as follows:

(1) Model Initialization: The central server initializes the model parameters and distributes them to the participating clients for training.

(2) Local Training: Each client independently trains the model on its own data and updates the model parameters. This step is entirely local, ensuring data privacy and security.

(3) Aggregation of Updates: The central server collects all the updates from the clients and aggregates them to improve the global model. There are various aggregation methods, with the most common being simple averaging of the updates.

(4) Repeat Iterations: Steps 1 to 3 are repeated until the model reaches the desired performance standards or the specified number of training rounds is completed.

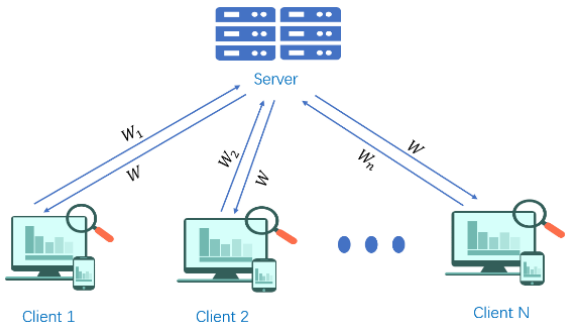


Figure 1. Horizontal Federated Learning Framework

A general federated learning system consists of a server and NNN clients. Each client k (k=1, 2, …, N) uses its own local dataset to train its local model parameters. The server's goal is to learn a model from the data of these clients and ultimately optimize the equation to convergence. However, model parameters may still contain a significant amount of privacy-sensitive information, and directly sharing them poses the risk of exposing user privacy.

4. FDL

4.1. Overall Framework

The main problem to be addressed in Federated Learning under localized differential privacy constraints is the impact of high-dimensional parameters on model performance. Specifically, the privacy budget is proportional to the dimensionality of the parameters, and as the parameter dimension increases, substantial noise must be introduced to protect privacy, resulting in larger model errors. Additionally, high-dimensional parameters also increase the communication burden of the model. To address these issues, this paper proposes a novel Federated Learning method under localized differential privacy, called FDL. This method evaluates the contribution of each dimension and uses the exponential mechanism to randomly select key dimensions, effectively reducing the total amount of noise required while retaining the core information of the data. The flowchart of FDL is shown in Figure 2.

The client trains the model using local data for the current round, generating updated model parameters;

Parameters with small values are discarded;

The FDL method is used to select the parameters;

Noise is added to the final selected parameters, generating perturbed parameters, which are then sent to the server for aggregation;

The server aggregates the client models.

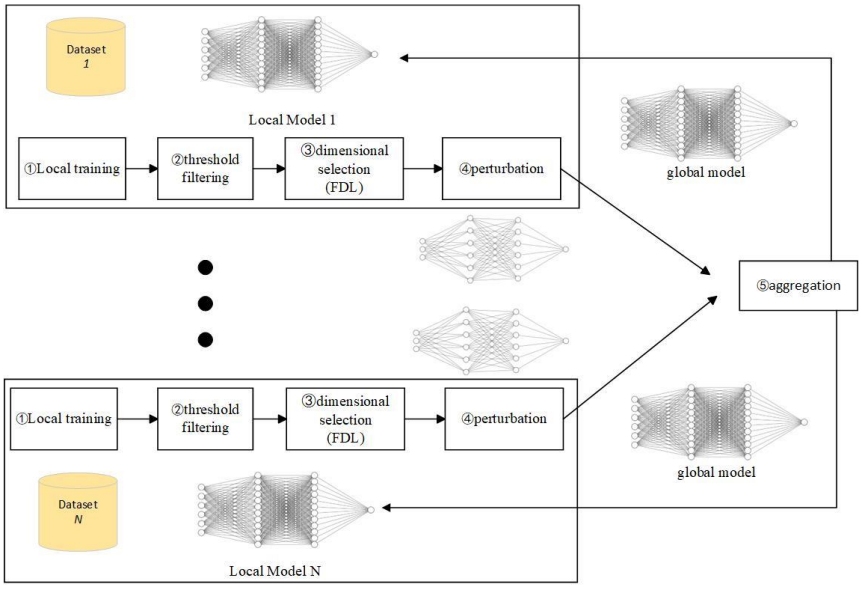


Figure 2. FDL Framework

4.2. FDL: Dimension Selection Based on the Exponential Mechanism

In the context of Federated Learning, training neural network models typically involves multiple iterations and generates a large number of parameters. Generally, most of the parameters or gradients

updated by the client are close to zero [27]. It is not a wise choice to use privacy budgets to protect these insignificant parameters. The client should focus on protecting those key parameters (i.e., values far from zero) to reduce the consumption of the privacy budget. However, simply protecting the parameters far from zero does not fully protect data privacy, as malicious attackers could infer that the unprotected parameters are close to zero. Therefore, an algorithm needs to be designed that can protect both the important parameters being disclosed and the privacy of the parameter selection process [19]. This implementation is described in Algorithm 1.

In Algorithm 1, we consider r as an adjustable real number approaching 0. To ensure the aggregation goal is achieved with the same communication cost, we introduce a threshold parameter th to control the number of selected parameters. The specific settings for rrr and thththh will be provided in the experimental section. First, we filter out the weight parameters that are close to 0, and then execute the th-based exponential selection mechanism. During the selection process, we set the utility function u to the absolute value of each parameter (1+|w|). Since the range of the original input data is between [−1, 1], the sensitivity of the utility function is $\Delta u=1$, and then parameters with higher contributions can be selected with a probability proportional to $e^{(\frac{(1+|w|)^2}{2})}$.

4.3. Privacy Analysis

Due to the nature of Federated Learning, external attackers can only indirectly access data by eavesdropping on the uploaded intermediate parameters or the downloaded aggregated results, which are obtained after being processed by local perturbation algorithms. According to the properties of differential privacy, as long as the local perturbation algorithm satisfies Local Differential Privacy, attackers cannot obtain any useful data from this information.

Theorem 2: The FDL mechanism satisfies ε−LDP.

Algorithm 1: FDL

Input: Client's parameter vector $\boldsymbol{w}_i \in [-1,1]^n$, threshold-filtered parameters r, privacy budget ε, selection quantity threshold th, filtered parameter vector $\boldsymbol{w}_s$

Output: Final selected parameter vector $\boldsymbol{w}_f$

1. Initialize r, th, and $\boldsymbol{w}_s$;
2. Normalize the parameters;
3. For each $w \in \boldsymbol{w}_i$ do:
4. If $w > r$ then
5. Add w to $\boldsymbol{w}_s$
6. end for
7. for $j = 1, 2, 3\ldots, th$ do:
8. Compute the utility function of the parameter u
9. Randomly select parameters from $\boldsymbol{w}_s$ without repetition with probability $e^{\frac{\varepsilon u}{2}}$, and add the selected parameters to $\boldsymbol{w}_f$.
10. end for

Proof: Given two adjacent parameter vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$, the utility function $u(\Delta u = 1)$, and the output range of the exponential mechanism, for any output element $\boldsymbol{o} \in \boldsymbol{O}$, the following equation holds:

$$
\begin{aligned}
&\frac{Pr[M_{EM}(\boldsymbol{w}, u, \boldsymbol{O}) = \boldsymbol{o}]}{Pr[M_{EM}(\boldsymbol{w}', u, \boldsymbol{O}) = \boldsymbol{o}]} \\
&= \frac{exp(\frac{\varepsilon u(\boldsymbol{w}, \boldsymbol{o})}{2})}{\sum_{\boldsymbol{o}' \in \boldsymbol{O}} \exp(\frac{\varepsilon u(\boldsymbol{w}, \boldsymbol{o}')}{2})} \Bigg/ \frac{exp(\frac{\varepsilon u(\boldsymbol{w}', \boldsymbol{o})}{2})}{\sum_{\boldsymbol{o}' \in \boldsymbol{O}} exp(\frac{\varepsilon u(\boldsymbol{w}', \boldsymbol{o}')}{2})} \\
&= \frac{exp(\frac{\varepsilon u(\boldsymbol{w}, \boldsymbol{o}')}{2})}{exp(\frac{\varepsilon u(\boldsymbol{w}', \boldsymbol{o})}{2})} \cdot \frac{\sum_{\boldsymbol{o}' \in \boldsymbol{O}} exp(\frac{\varepsilon u(\boldsymbol{w}', \boldsymbol{o}')}{2})}{\sum_{\boldsymbol{o}' \in \boldsymbol{O}} exp(\frac{\varepsilon u(\boldsymbol{w}, \boldsymbol{o}')}{2})} \\
&\le exp(\frac{\varepsilon}{2}) \cdot exp(\frac{\varepsilon}{2}) \cdot \frac{\sum_{\boldsymbol{o}' \in \boldsymbol{O}} exp(\frac{\varepsilon u(\boldsymbol{w}, \boldsymbol{o}')}{2})}{\sum_{\boldsymbol{o}' \in \boldsymbol{O}} exp(\frac{\varepsilon u(\boldsymbol{w}, \boldsymbol{o}')}{2})}
\end{aligned}
\tag{5}
$$

5. EXPERIMENTS

5.1. Setup

The experiments were conducted using Python and PyTorch 1.11 to simulate the proposed scheme on a single GPU configuration. The detailed experimental environment included an Ubuntu 20.04 operating system, an Intel(R) Core (TM) i9-14900KF@6.0GHz CPU, an NVIDIA GeForce RTX 4080 GPU, 32GB of RAM, and a 1TB SSD.

The MNIST [28] dataset was selected for the experiments. This dataset contains 10 classes of handwritten digits, with 60,000 training samples and 10,000 test samples. Both IID and Non-IID data distribution scenarios were considered. The convolutional neural network used included an input layer, two convolutional layers, two max-pooling layers, and two fully connected layers, consistent with the CNN architectures in [3]. The local SGD algorithm used a batch size of 64, a learning rate of 0.01, and 100 clients.

This paper compared four algorithms: LDP-FL [20], NoLDP-FL [3], SignDS-FL [29], and FDL. Each algorithm was run 10 times, and the average results were reported to ensure stability and reliability.

5.2. Experimental Analysis

5.2.1. Communication Cost Analysis

As described in Algorithm 1, different values of r and th affect the model's accuracy. Therefore, for r and th, the experiment followed the setup in [26], setting r to 0.01, th to 0.8 of the number of filtered parameters, the privacy budget of the Laplace mechanism to ε=0.5. The experimental results are shown in Table 1.

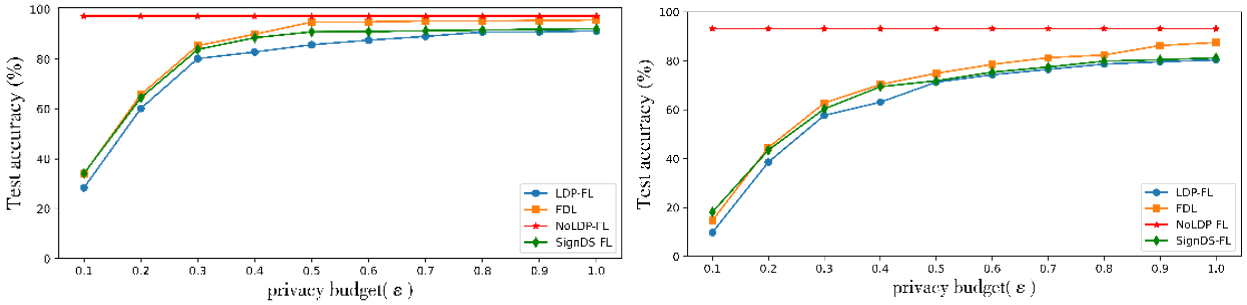
Where C represents the total number of parameters, C_r represents the number of parameters filtered by the threshold, C_s represents the number of parameters finally selected, and ACC represents the testing accuracy. The results indicate that this scheme can save at least 60% of the communication cost.

Table 1. The Impact of r on Model Performance

r	C_r	C_s	Acc
0.015	0.22C	0.16C	53.25%
0.012	0.35C	0.28C	74.50%
0.010	0.47C	0.38C	94.83%
0.008	0.58C	0.46C	87.46%

5.2.2. The Impact of Different Privacy Budgets on Model Performance

The size of the privacy budget indicates the level of privacy protection. As shown in Figure 3, as the privacy budget increases, the noise gradually decreases, and the model performance progressively improves. On the MNIST dataset, with the increase of the privacy budget, the test accuracy of the FDL algorithm surpasses that of other algorithms, thereby enhancing the model's accuracy.

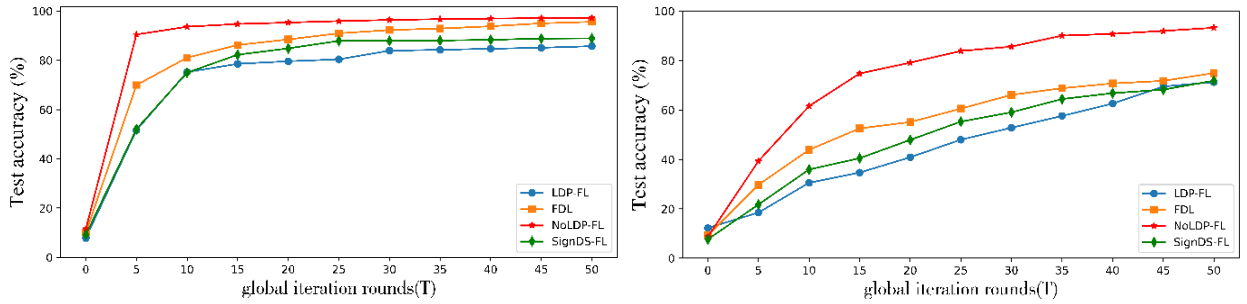


a) MNIST-IID b) MNIST-NOIID

Figure 3. The Impact of Different Privacy Budgets on Model Performance

5.2.3. The Impact of Global Iteration Count (T) on Model Performance

As training progresses, the model becomes better adapted to the data distribution. However, excessive iterations may lead to overfitting, increase computational costs, and reduce model accuracy. This section's experiments investigate the impact of the number of iterations on model accuracy. In these experiments, the number of communication rounds is set to 50, the number of participants is 100, and the privacy budget ranges from 0.1 to 1. Through testing, the proposed method in this paper demonstrates advantages over related work. As shown in Figure 4, model accuracy improves with an increasing number of iterations and tends to converge around 50 rounds, as depicted in Figure 4(a). Figure 4(b) illustrates that in Non-IID scenarios, the impact of iteration count is more significant.



a) MNIST-IID b) MNIST-NOIID

Figure 4. The Impact of Global Iteration Count (T) on Model Performance

6. CONCLUSION

This paper addresses the issue of privacy budget explosion caused by handling high-dimensional data in differentially private federated learning and thus proposes a dimension selection algorithm. In the

dimension selection algorithm, the utility score of each parameter is calculated based on the magnitude of the parameter to measure its importance. Finally, comparative simulations on the MNIST dataset demonstrate the effectiveness of the proposed approach.

REFERENCES

[1] WANG X, LI J, LI J, et al. Multilevel similarity model for high-resolution remote sensing image registration [J]. Information Sciences, 2019, 505: 294-305.

[2] CAI Y, LUAN T, GAO H, et al. YOLOv4-5D: An effective and efficient object detector for autonomous driving [J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-13.

[3] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [C]// Artificial Intelligence and Statistics. PMLR, 2017: 1273-1282.

[4] MELIS L, SONG C, DE CRISTOFARO E, et al. Exploiting unintended feature leakage in collaborative learning [C]// 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019: 691-706.

[5] NASR M, SHOKRI R, HOUMANSADR A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning [C]// 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019: 739-753.

[6] XU G, LI H W, ZHANG Y, et al. Privacy-preserving federated deep learning with irregular users [J]. IEEE Transactions on Dependable and Secure Computing, 2020, 19 (2): 1364-1381.

[7] TRUEX S, BARACALDO N, ANWAR A, et al. A hybrid approach to privacy preserving federated learning [C]// Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security,New York, NY, USA, 2019: 1-11.

[8] YIN, L, FENG J, XUN H, et al. A privacy-preserving federated learning for multiparty data sharing in social IoTs [J]. IEEE Transactions on Network Science and Engineering, 2021,8(3), 2706–2718.

[9] DWORK C. Differential privacy [C] //Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, 2006. Berlin, Heidelberg: Springer, 2006, Part II 33: 1-12.

[10] YANG M M, GUO T L, ZHU T Q, et al. Local differential privacy and its applications: A comprehensive survey [J]. Computer Standards & Interfaces, 2024, 89: 103827.

[11] ZHAO Y, ZHAO J, YANG M, et al. Local differential privacy based federated learning for internet of things [J]. IEEE Internet of Things Journal, 2021, 8(11):8836–8853.

[12] JIA W, SUN M, LIAN J, et al. Feature dimensionality reduction: a review [J]. Complex & Intelligent Systems, 2022, 8(3): 2663-2693.

[13] MAHAWAGA ARACHCHIGE P C, BERTOK P, KHALIL I, et al. Local differential privacy for deep learning [J]. IEEE Internet Things, 2020, 7(7): 5827–5842.

[14] WANG, N, Xiao X K, YANG Y, et al. Collecting and analyzing multidimensional data with local differential privacy[C]//2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 2019: 638-649.

[15] WANG Y S, TONG Y X, SHI D Y. Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework [J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4):6283-6290.

[16] Shin, H., Kim, S., Shin, J., & Xiao, X. Privacy enhanced matrix factorization for recommendation with local differential privacy [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30:1770–1782.

[17] Wang J Z, Kong L W, Huang Z, et al. Research review of federated learning algorithms [J]. Big Data Research, 2020, 6(6): 64-82.

[18] Wang B, Yao Y, Shan S, et al. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks [C]//2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019: 707-723.

[19] SHOKRI R, SHMATIKOV V. Privacy-preserving deep learning [C]// Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2015: 1310-1321.

[20] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy [C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Ne-w York, NY, USA, 2016: 308-318.

[21] GEYER R C, KLEIN T, NABI M. Differentially private federated learning: A client level perspective [J]. arXiv preprint arXiv:1712.07557, 2017.

[22] SHIN H, KIM S, SHIN J, et al. Privacy enhanced matrix factorization for recommendation with local differential privacy [J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1770-1782.

[23] MIAO Y, XIE R, LI X, et al. Compressed federated learning based on adaptive local differential privacy[C]//Proceedings of the 38th Annual Computer Security Applications Conference, New York, NY, USA, 2022: 159-170.

[24] CUI L, MA J, ZHOU Y, et al. Boosting accuracy of differentially private federated learning in industrial IoT with sparse responses [J]. IEEE Transactions on Industrial Informatics, 2022, 19(1): 910-920.

[25] WANG B, CHEN Y, JIANG H, et al. PPeFL: Privacypreserving edge federated learning with local differential privacy [J]. IEEE Internet of Things Journal, 2023.

[26] Liu R, CAO Y, YOSHIKAWA M, et al. Fedsel: Federated sgd under local differential privacy with top-k dimension selection[C]//Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, 2020, Proceedings, Part I 25. Springer International Publishing, 2020: 485-501.

[27] RASHID T. Make Your Own Neural Network [M]. CreateSpace Independent Publishing Platform, 2016.

[28] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[29] JIANG X, ZHOU X, GROSSKLAGS J. Sign-fl: Local differentially private federated learning with sign-based dimension selection [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2022, 13(5): 1-22.