EasyChair Preprint

№ 12740

DPLE: a Privacy-Enhanced and Straggler-Resilient Distributed Learning Framework for Smart Cloud

Yilei Xue, Jianhua Li and Jun Wu

March 27, 2024

DPLE: A Privacy-Enhanced and Straggler-Resilient Distributed Learning Framework for Smart Cloud

1st Yilei Xue
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
xueyilei@sjtu.edu.cn

2nd Jianhua Li
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
lijh888@sjtu.edu.cn

3rd Jun Wu
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
jun.wu@ieee.org

***Abstract*—In the smart cloud environment, distributed learning faces privacy and straggler issues. Lagrange coded computing can alleviate these concerns to some extent. However, when the number of honest but curious nodes exceeds a certain threshold, or there exists outside eavesdroppers, the privacy of the system will be threatened. To address this challenge, we propose a differentially private Lagrange encoding distributed learning framework, named DPLE. Firstly, we utilize Lagrange encoding to hide the raw data and inject redundancy, thereby enhancing privacy protection and resilience against stragglers. Additionally, artificial noise will be injected into local computation results, further securing sensitive information against leakage. Moreover, we conduct theoretical analyses to determine the variance of artificial noise required to achieve a certain level of privacy protection within this framework. Through experiments, we validate the effectiveness of the proposed framework and assess the influence of various system parameter settings on accuracy.**

***Index Terms*—Lagrange coded computing, differential privacy, distributed learning, artificial noise**

I. Introduction

Cloud computing infrastructure enables distributed machine learning to effectively manage extensive datasets and complex models [1]. However, several unresolved challenges crucial for advancing distributed learning in smart cloud environments persist, including issues related to performance, fault tolerance, portability, and privacy concerns [2]. Our paper primarily focuses on the privacy issues associated with distributed learning. This is particularly pertinent in cloud environments where physical infrastructure is shared, with multiple users utilizing the same host and being partitioned by virtual machines [3]. While this approach facilitates scalability in cloud systems, it also introduces significant security and privacy risks. When semi-honest nodes attempt to acquire sensitive information or adversaries and aim to eavesdrop on the training outcomes, the privacy of the system will be compromised. In response to the aforementioned privacy threats, cryptographic techniques such as homomorphic encryption [4] and secure multiparty computation [5] have been widely employed. However, cryptographic techniques require significant computational overhead.

Recently, the combination of coding techniques and distributed learning has gained increasing attention as a way to safeguard the system's privacy [6]. For example, Coded Federated Learning (CFL) leverages coding technique and federated learning to mitigate privacy and straggler issues [7]. Additionally, Lagrange coded computing (LCC) utilizes Lagrange interpolation polynomials to encode datasets, ensuring an optimal balance among privacy, security, and resiliency [8]. However, LCC is limited to handling multivariate polynomial functions of the input dataset. To address this limitation, CodedPrivateML [3] enhances LCC by integrating an approximation of the sigmoid function, thereby facilitating logistic regression within the Lagrange coded computing framework. Nonetheless, relying solely on Lagrange interpolation polynomials offers limited protection for system privacy. When honest-but-curious nodes exceed the system's capacity or when adversaries attempt to eavesdrop on the system, the privacy of the system remains vulnerable.

One effective way to enhance the system's privacy protection is by introducing artificial noise, with differential privacy (DP) being a typical example. Differential privacy [9] offers statistical information without revealing individual data, thereby ensuring the protection of system privacy. Numerous studies have already delved into the application of differential privacy in distributed learning. For instance, in [10], DP is seamlessly integrated into federated learning, and the required variance of artificial noise for a specific privacy protection level is quantitatively analyzed. Additionally, [11] proposes a privacy-preserving method for learning personalized models on distributed data while ensuring differential privacy. Moreover, in [12], the authors combine DP and homomorphic encryption in federated learning using a novel stochastic quantization operator. In this paper, we further extend the application of differential privacy by deploying it into the LCC framework to enhance the privacy protection level.

We focus on a scenario where a master exclusively possesses the dataset and aims to distribute computationally intensive tasks to multiple workers while preserving privacy. To achieve this, we delve into coding and differential privacy. Specifically, we utilize Lagrange interpolation polynomials to encode the dataset, thereby facilitating privacy protection and injecting redundancy. Besides, after local computation, artificial noise will be introduced to local results to further safeguard the system's privacy. This system not only combats eavesdropping concerns from honest but curious nodes and

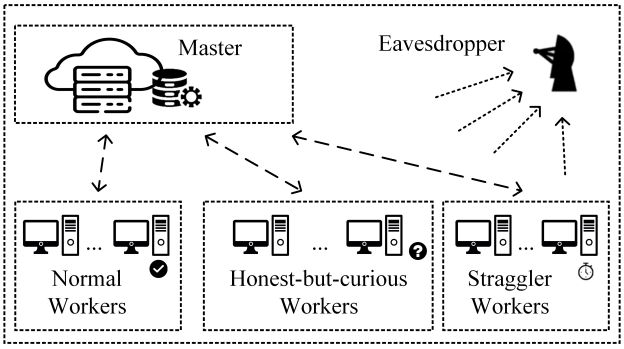

Fig. 1. The scenario of DPLE.

outside adversaries but also mitigates the straggler effect to some extent. Additionally, we provide the variance of the noise required for specific privacy protection levels and validate the system's accuracy through experiments.

II. System Architecture

As illustrated in Fig. 1, we consider a master-worker architecture, comprising a master and N workers. The master aims to train a logistic regression model by distributing computationally intensive tasks to multiple workers without compromising privacy. However, during the training process, there may be instances of honest but curious workers attempting to obtain sensitive information illegally, stragglers unable to complete computations in a timely manner, and outside adversaries eavesdropping on the training results. To ensure the privacy of the system, we propose a differentially private Lagrange encoding framework, namely DPLE. Within this framework, the master possesses the dataset $\mathbf{X} \in \mathbb{R}^{m\times d}$ along with the label vector $\mathbf{y} \in \{0,1\}^{m\times 1}$, where m and d denote the number of data items and the number of features, respectively. The model weight $\mathbf{w}$ is derived by minimizing the cross-entropy loss function $L(\mathbf{w})$, with its gradient denoted as $\nabla L(\mathbf{w}) = \frac{1}{m}\mathbf{X}^\top(S(\mathbf{X}\cdot\mathbf{w}) - \mathbf{y})$. Here, $S(\cdot)$ represents the element-wise application of the sigmoid function. Then, in the t-th iteration, the updated weight is represented as $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \frac{\eta}{m}\mathbf{X}^\top(S(\mathbf{X}\mathbf{w}^{(t)}) - \mathbf{y})$, where η denotes the learning rate.

The system architecture of DPLE is presented in Fig. 2. Before the master distributes the data to the workers for computation, it needs to encode the data using Lagrange interpolation polynomials. Therefore, the master first normalize the dataset $\mathbf{X}$ into a normalized dataset $\bar{\mathbf{X}}$ with zero mean and unit variance, and then $\bar{\mathbf{X}}$ is evenly partitioned into K shares $\bar{\mathbf{X}} = [\bar{\mathbf{X}}_1^T, \ldots, \bar{\mathbf{X}}_K^T]^T$ with $\bar{\mathbf{X}}_i \in \mathbb{R}_q^{\frac{m}{K}\times d}$ $(i = 1, \ldots, K)$ being the i-th share. We assume that the number of entries m is evenly divisible by K. If not, any remaining entries can be disregarded. Then, the encoded dataset $\hat{\mathbf{X}}_i$ $(i = 1, \ldots, N)$ can be represented as $\hat{\mathbf{X}}_i = g_x(m_i)$ with $g_x(\beta)$ being

$$g_x(\beta) = \sum_{i=1}^{K} \bar{\mathbf{X}}_i \prod_{k\neq i} \frac{\beta - u_k}{u_i - u_k} + \sum_{i=K+1}^{K+T} \mathbf{Z}_i \prod_{k\neq i} \frac{\beta - u_k}{u_i - u_k}. \quad (1)$$

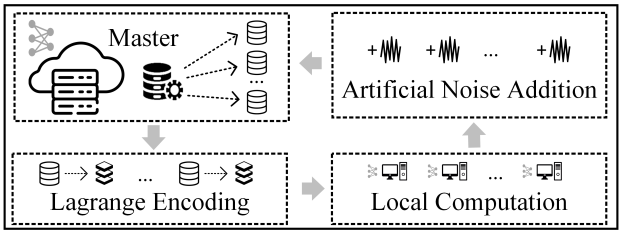

Fig. 2. System architecture of DPLE.

In Eq.(1), $\mathbf{Z}i \in \mathbb{R}q^{\frac{m}{K}\times d}$ is the redundancy matrix with elements uniformly distributed within the range $[-\theta, \theta]$ for some $\theta \in \mathbb{R}$. T represents the maximum number of colluding nodes the system can tolerate without compromising privacy. Additionally, $u_1, \ldots, u_{K+T}$ denote $(K+T)$ distinct numbers chosen by the master, and $\{m_i\}_{i=1}^N$ represent another N distinct numbers selected by the master, which are different from $u_1, \ldots, u_{K+T}$. That is $\{u_\alpha\}_{\alpha=1}^{K+T} \cap \{m_i\}_{i=1}^N = \varnothing$. Similarly, the master employs the identical set of $\{m_i\}_{i=1}^N$ for encoding the weight vector $\mathbf{w}^{(t)}$, and the i-th encoded weight vector is $\hat{\mathbf{w}}_i^{(t)} = g_w(m_i)$ with $g_w(m_i)$ being

$$g_w(\beta) = \sum_{i=1}^{K} \mathbf{w}^{(t)} \prod_{k\neq i} \frac{\beta - u_k}{u_i - u_k} + \sum_{i=K+1}^{K+T} \mathbf{v}_i \prod_{k\neq i} \frac{\beta - u_k}{u_i - u_k}. \quad (2)$$

The entries within $\mathbf{v}_i \in \mathbb{R}_q^{d\times r}$ are randomly chosen with uniform distribution from the range $[-\xi, \xi]$ for some $\xi \in \mathbb{R}$. To achieve more accurate results for the master's recovery, it is suggested that $\{u_\alpha\}$ and $\{m_i\}$ are close to each other and $\min\{m_i\} < \{u_\alpha\}_{\alpha=1}^{K+T} < \max\{m_i\}$ for $i = 1, \ldots, N$. It is evident that both $g_x(\beta)$ and $g_w(\beta)$ have a degree of $(K+T-1)$.

Following the encoding process, the master distributes the encoded dataset $\hat{\mathbf{X}}_i$ and weight vector $\hat{\mathbf{w}}_i^{(t)}$ to the i-th worker for computation. Given that Lagrange coding is intended for polynomial computation, we opt for an r-th order approximation of the sigmoid function $\hat{S}(z) = \sum_{i=0}^{r} a_i z^i$ with a_i being estimated according to the least squares estimation of the sigmoid function. Then, the i-th $(i = 1, \ldots, N)$ worker computes function f locally with f being

$$f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)}) = \hat{\mathbf{X}}_i^\top \cdot \hat{S}(\hat{\mathbf{X}}_i \cdot \hat{\mathbf{w}}_i^{(t)}). \quad (3)$$

Then, we can deduce that the degree of $f(\cdot)$ is $(2r+1)$, indicating that a total of $(2r+1)$ points are required to reconstruct $f(\cdot)$. In the t-th iteration, once the i-th worker completes computation, the computed result $f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)})$ is added with artificial noise $\mathbf{n}_i^{(t)} \in \mathbb{R}_q^{d\times 1}$ by worker i to satisfy DP. Consequently, the perturbed result $\tilde{f}_i^{(t)}$ is expressed as

$$\tilde{f}_i^{(t)} = \hat{\mathbf{X}}_i^\top \cdot \hat{S}(\hat{\mathbf{X}}_i \cdot \hat{\mathbf{w}}_i^{(t)}) + \mathbf{n}_i^{(t)}. \quad (4)$$

Then, the i-th worker sends the noised result $\tilde{f}_i^{(t)}$ to the master.

After receiving $\tilde{f}_i^{(t)}$, the master utilizes the received results $\tilde{f}_i^{(t)}$ $(i \in |\mathcal{D}|)$ to construct a new polynomial $\tilde{h}(\beta)$ at the t-th iteration with $\mathcal{D}$ representing the set of the first $|\mathcal{D}| = (2r+1)(K+T-1)+1$ workers who have completed the computing tasks. It should be noted that $N \geq (2r+1)(K+T-1)+S_s+1$

with S_s being the number of stragglers the system can withstand. The new polynomial $\tilde{h}(\beta)$ is constructed by executing Lagrange interpolation on the pairs $(m_i, \tilde{h}(m_i))$ with

$$\tilde{h}(m_i) \triangleq \tilde{f}_i^{(t)} = f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)}) + \mathbf{n}_i^{(t)}. \tag{5}$$

Hence the function values at u_α $(\alpha = 1, \ldots, K)$ can be expressed as

$$\tilde{h}(u_\alpha) = \sum_{i\in\mathcal{D}} \tilde{f}_i^{(t)} \cdot \prod_{j\in\mathcal{D}\setminus\{i\}} \frac{u_\alpha - m_j}{m_i - m_j}. \tag{6}$$

Then, substituting Eq. (4) into Eq. (6), we can get

$$\tilde{h}(u_\alpha) = \sum_{i\in\mathcal{D}} [\hat{\mathbf{X}}_i^\top \hat{S}(\hat{\mathbf{X}}_i \cdot \hat{\mathbf{w}}_i^{(t)}) + \mathbf{n}_i^{(t)}] \cdot \prod_{j\in\mathcal{D}\setminus\{i\}} \frac{u_\alpha - m_j}{m_i - m_j} \tag{7}$$

And we define the equivalent noise $\tilde{\mathbf{n}}_\alpha^{(t)}$ $(\alpha = 1, \ldots, K)$ as

$$\tilde{\mathbf{n}}_\alpha^{(t)} \triangleq \sum_{i\in\mathcal{D}} \mathbf{n}_i^{(t)} \cdot \prod_{j\in\mathcal{D}\setminus\{i\}} \frac{u_\alpha - m_j}{m_i - m_j}. \tag{8}$$

Then we know

$$\tilde{h}(u_\alpha) = h(u_\alpha) + \tilde{\mathbf{n}}_\alpha^{(t)}, \tag{9}$$

where

$$h(u_\alpha) = \sum_{i\in\mathcal{D}} f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)}) \cdot \prod_{j\in\mathcal{D}\setminus\{i\}} \frac{u_\alpha - m_j}{m_i - m_j}. \tag{10}$$

After obtaining the function value of $\tilde{h}(u_\alpha)$ $(\alpha = 1, \ldots, K)$, the master aggregates the K decoded results to acquire

$$\sum_{\alpha=1}^{K} \tilde{h}(u_\alpha) = \sum_{\alpha=1}^{K} [f(\bar{\mathbf{X}}_\alpha, \mathbf{w}^{(t)}) + \tilde{\mathbf{n}}_\alpha^{(t)}], \tag{11}$$

where $\sum_{\alpha=1}^{K} f(\bar{\mathbf{X}}_\alpha, \mathbf{w}^{(t)}) = \bar{\mathbf{X}}^\top \hat{S}(\bar{\mathbf{X}} \cdot \mathbf{w}^{(t)})$. Next, the master updates the gradient based on

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \frac{\eta}{m}\{\bar{\mathbf{X}}^\top[(\hat{S}(\bar{\mathbf{X}} \cdot \mathbf{w}^{(t)}) - \mathbf{y})] + \sum_{\alpha=1}^{K} \tilde{\mathbf{n}}_\alpha^{(t)}\}. \tag{12}$$

This can alternatively be written as

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \frac{\eta}{m}[\sum_{\alpha=1}^{K} \tilde{h}(u_\alpha) - \bar{\mathbf{X}}^\top \mathbf{y}]. \tag{13}$$

III. Analysis of Artificial Noise Variance

We now discuss the requirements that the variance of the artificial noise $\mathbf{n}_i^{(t)}$ added by the i-th worker needs to satisfy to attain a specific level of privacy protection. We consider the situation where the added artificial noise follows a Gaussian distribution. Firstly, the definition of differential privacy is provided as follows:

Definition 1 ((ε, δ)-DP). A randomized mechanism $\mathcal{M}$ operating on a domain $\mathbb{N}$ is said to be (ε, δ)-differentially private if for any two adjacent databases D, $D^{'} \in \mathbb{N}$, and $\forall \mathcal{S} \subseteq \text{Range}(\mathcal{M})$:

$$\Pr(\mathcal{M}(D) \in \mathcal{S}) \leq e^{\varepsilon} \Pr(\mathcal{M}(D^{'}) \in \mathcal{S}) + \delta. \tag{14}$$

The Gaussian mechanism of differential privacy is satisfied when $\mathcal{M}(D) = L(D) + n$ with n following Gaussian distribution $N(0, \sigma^2)$ with mean 0 and variance σ^2. To ensure that $\mathcal{M}$ adheres to (ε, δ)-DP, it is required that $\sigma^2 \geq \frac{2\ln(1.25/\delta)\Delta_{\text{Gau}}^2}{\varepsilon^2}$,

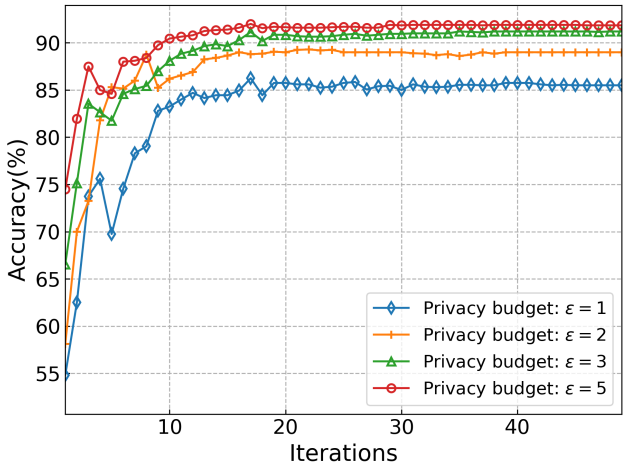

Fig. 3. The impact of different privacy budgets on accuracy.

where ΔGau denotes the sensitivity of the Gaussian mechanism, calculated as $\Delta_{\text{Gau}} = \max_{D,D'} \|L(D') - L(D)\|_2$.

For the i-th worker in the t-th iteration, we define a matrix $\mathbf{R}^{(t),i} \triangleq \hat{\mathbf{X}}_i^\top \odot (\mathbf{1}_d \cdot [\hat{S}(\hat{\mathbf{X}}_i \cdot \hat{\mathbf{w}}_i^{(t)})]^\top)$, where $\odot$ denotes the Hadamard product and $\mathbf{1}_d$ represents a column vector with all d elements being 1. Then we can derive that $f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)}) = \sum_{\varsigma=1}^{m/K} \mathbf{R}_\varsigma^{(t),i}$ with $\mathbf{R}_\varsigma^{(t),i}$ being the ς-th column of $\mathbf{R}^{(t),i}$. We assume that $\mathbf{R}_\varsigma^{(t),i}$ is bounded by $||\mathbf{R}_\varsigma^{(t),i}||_2 \leq B_2^{(t),i}$. Then, the sensitivity of the i-th worker can be written as $\Delta\text{Gau} = \max_{\hat{\mathbf{X}}_i, \hat{\mathbf{X}}_i'} ||f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)}) - f(\hat{\mathbf{X}}_i', \hat{\mathbf{w}}_i^{(t)})||_2$, where $f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)})$ and $f(\hat{\mathbf{X}}_i', \hat{\mathbf{w}}_i^{(t)})$ respectively represent the function value for the neighboring dataset $\hat{\mathbf{X}}_i$ and $\hat{\mathbf{X}}_i'$ at the t-th iteration. According to Eq. (3), we know that a single data change can cause the computed result $f(\hat{\mathbf{X}}_i, \hat{\mathbf{w}}_i^{(t)})$ to vary by up to $2B_2^{(t),i}$, which means $\Delta\text{Gau} = 2B_2^{(t),i}$. Then, for the i-th worker, in the t-th iteration, if the noise $\mathbf{n}_i^{(t)} \sim \mathcal{N}(0, \sigma_{(t),i}^2)$, the $(\varepsilon_i, \delta_i)$-DP can be guaranteed when $\sigma_{(t),i}$ satisfies

$$\sigma_{(t),i} \geq \sqrt{2\ln(1.25/\delta_i)} \cdot 2B_2^{(t),i}/\varepsilon_i. \tag{15}$$

IV. Experiment

In this section, we demonstrate the effectiveness of the proposed DPLE framework through experimental verification. We conduct logistic regression training using the MNIST and FashionMNIST datasets. For the MNIST dataset, we select a total of 12,700 samples from two classes with labels 1 and 2 for training. As for the FashionMNIST dataset, we choose two classes, 'Pullover' and 'Dress', for binary classification training. Each sample in these two datasets has $28 \times 28 + 1 = 785$ features, where the additional feature count is due to the inclusion of a bias term. The privacy budget for each worker is uniformly expressed as ε, and the relaxation term δ is set to 0.01.

Fig. 3 presents the training accuracy of the Fashion-MNIST dataset under varying privacy budgets while keeping other parameters fixed. The initial parameter configuration here is

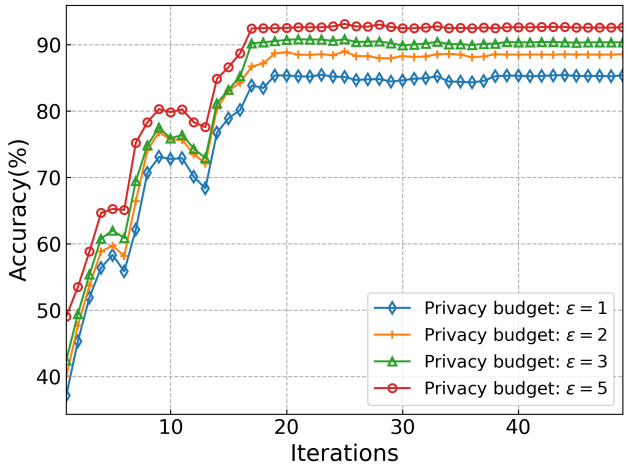

Fig. 4. The impact of different privacy budgets on accuracy (MNIST).

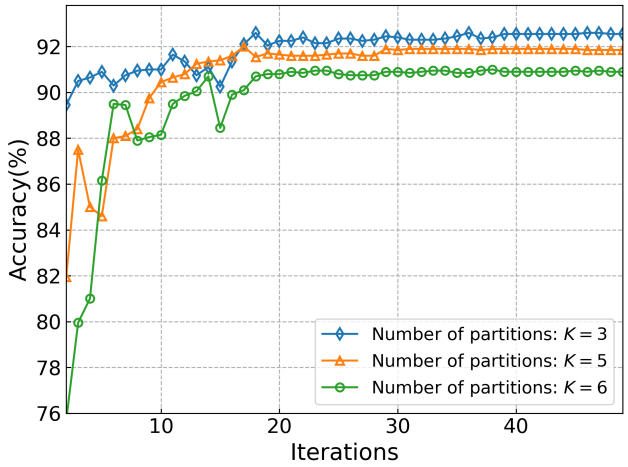

Fig. 5. The influence of different dataset partitions on Accuracy (Fashion-MNIST).

$N = 50$, $K = 5$, and $T = 4$. Since it can be calculate that $|\mathcal{D}| = 3 \times (5+4-1)+1 = 25$, then we know the system can accommodate $N - |\mathcal{D}| = 25$ stragglers. It can be observed from Fig. 3 that as the privacy budget decreases, the training accuracy of the dataset gradually decreases, while the privacy protection level of the system gradually increases. This trend arises because a higher privacy protection level necessitates greater artificial noise variance, which consequently affects the accuracy of the system's training. Thus, there exists a trade-off between the privacy protection level and the training accuracy.

In Fig. 4, we present the training accuracy of the MNIST dataset under different privacy budgets. The parameter configuration here is $N = 50$, $K = 5$, and $T = 2$. It can be observed that the training accuracy gradually increases as the privacy budget increases; however, a larger privacy budget corresponds to a lower level of privacy protection. This observation is consistent with the analysis presented in Fig. 3.

Fig. 5 demonstrates how different dataset partitions affect training accuracy. Here, the parameter settings are $N = 50$, $T = 4$, and $\varepsilon = 5$. It's noticeable that as the number of dataset partitions increases, training accuracy gradually declines. This is related to the fact that fewer dataset partitions allow each encoded dataset to contain richer information. While fewer partitions are beneficial for training accuracy, too few may impose excessive computational burdens on individual workers. Therefore, it is necessary to select an appropriate number of dataset partitions to balance the computation load of individual workers and the overall training accuracy.

V. Conclusion

To address the threats posed by an excessive number of honest but curious nodes and external eavesdroppers in the LCC framework, we introduced differential privacy on top of the existing LCC framework. Firstly, we employed Lagrange interpolation polynomials for encoding to safeguard sensitive data from potential leaks. Then we introduced artificial noise to further enhance the protection of sensitive information within the LCC framework. Subsequently, we analyzed the required magnitude of noise variance for achieving specific privacy protection levels. Through experimentation, we validated the effectiveness of our proposed solution and explored how various system parameters affect training accuracy.

VI. Acknowledgment

This work was supported in part by the JSPS KAKENHI under Grants 23K11072, and in part by the National Natural Science Foundation of China under Grants U21B2019.

References

[1] D. Pop, "Machine learning and cloud computing: Survey of distributed and saas solutions," *arXiv preprint arXiv:1603.08767*, 2016.

[2] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *Acm computing surveys (csur)*, vol. 53, no. 2, pp. 1–33, 2020.

[3] J. So, B. Güler, and A. S. Avestimehr, "Codedprivateml: A fast and privacy-preserving framework for distributed machine learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 441–451, 2021.

[4] X. Yi, R. Paulet, E. Bertino, X. Yi, R. Paulet, and E. Bertino, *Homomorphic encryption*. Springer, 2014.

[5] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[6] J. S. Ng, W. Y. B. Lim, N. C. Luong, Z. Xiong, A. Asheralieva, D. Niyato, C. Leung, and C. Miao, "A comprehensive survey on coded distributed computing: Fundamentals, challenges, and networking applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1800–1837, 2021.

[7] S. Dhakal, S. Prakash, Y. Yona, S. Talwar, and N. Himayat, "Coded federated learning," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[8] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1215–1225.

[9] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[10] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[11] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[12] A. G. Sébert, M. Checri, O. Stan, R. Sirdey, and C. Gouy-Pailler, "Combining homomorphic encryption and differential privacy in federated learning," in *2023 20th Annual International Conference on Privacy, Security and Trust (PST)*, 2023, pp. 1–7.