Journal of The Korea Institute of Information Security & Cryptology
VOL.34, NO.1, Feb. 2024

ISSN 1598-3986(Print)
ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.1.143

연합 학습 환경에서의 Task-Specific Adaptive Differential Privacy 메커니즘 평가 방안 연구*

우타리예바 아셈,[1†] 최 윤 호[2‡]
[1,2]부산대학교 (대학원생, 교수)

Study on Evaluation Method of Task-Specific Adaptive Differential Privacy Mechanism in Federated Learning Environment*

Assem Utaliyeva,[1†] Yoon-Ho Choi[2‡]
[1,2]Pusan National University (Graduate student, Professor)

요 약

연합 학습(FL)은 여러 공동 작업자 간에 분산된 모델 학습을 위한 강력한 방법론으로 부상해 데이터 공유의 필요성을 없애준다. FL은 데이터 프라이버시를 보호하는 기능으로 호평을 받고 있지만, 다양한 유형의 프라이버시 공격으로부터 자유롭지 않다. 대표적인 개인정보 보호 기술인 차분 프라이버시(DP)는 이러한 취약점에 대응하기 위해 널리 사용된다. 이 논문에서는 기존의 작업별 적응형 DP 메커니즘을 FL 환경에 적용해 성능을 평가한다. 포괄적인 분석을 통해 다양한 DP 메커니즘이 공유 글로벌 모델의 성능에 미치는 영향을 평가하며, 특히 다양한 데이터 배포 및 분할 스키마에 주의를 기울인다. 이를 통해, FL에서 개인정보 보호와 유용성 간의 복잡한 상호 작용에 대한 이해를 심화하고, 성능 저하 없이 데이터를 보호할 수 있는 검증된 방법론을 제공한다.

ABSTRACT

Federated Learning (FL) has emerged as a potent methodology for decentralized model training across multiple collaborators, eliminating the need for data sharing. Although FL is lauded for its capacity to preserve data privacy, it is not impervious to various types of privacy attacks. Differential Privacy (DP), recognized as the golden standard in privacy-preservation techniques, is widely employed to counteract these vulnerabilities. This paper makes a specific contribution by applying an existing, task-specific adaptive DP mechanism to the FL environment. Our comprehensive analysis evaluates the impact of this mechanism on the performance of a shared global model, with particular attention to varying data distribution and partitioning schemes. This study deepens the understanding of the complex interplay between privacy and utility in FL, providing a validated methodology for securing data without compromising performance.

Keywords: Differential Privacy, Federated Learning, Adaptive Differential Privacy

Received(11. 16. 2023), Modified(01. 22. 2024), Accepted(01. 26. 2024)

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2022R1I1A3055233) and National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2023-00217689) and MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2023-RS-2023-00259967) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

† 주저자, assemakx@pusan.ac.kr
‡ 교신저자, yhchoi@pusan.ac.kr(Corresponding author)

I. Introduction

Federated Learning (FL) is gathering recognition as efficient way of decentralized training of a model among multiple collaborators without the need to share their data. In practice, FL is actually deployed in mobile devices [1][2], smart healthcare [3][4], financial services [5][6] and in other fields.

Usually, the concept of FL is also used in the context of data privacy and security, given that the train data remains in the local device and thus reduces the possibility of data leakages.

However, multiple studies proved that data in FL environment is still vulnerable to privacy attacks, such as GAN attack introduced in [7], or reconstructing the train data from gradients [8], and other attacks [9][10].

To address this issue, traditional privacy-preservation techniques such as differential privacy (DP) [11][12], blockchain [13], secure multi-party computation (SMPC) [14][15] are also being actively studied as a defense method against privacy attacks.

Specifically, we focus on DP, that is currently recognized as a golden standard of privacy-preservation techniques. The concept behind DP is to guarantee privacy by addition of carefully calibrated random noise directly to the input data, generated output or to the model hyperparameters. Generation of differentially private data by addition of the random noise directly to the data is the most straightforward and one of the strongest ways to protect the data, while at the same time, it introduces the largest utility and performance degradation. Recently, researchers are focused on methods to resolve the above-mentioned privacy and utility tradeoff problem. One of such methods is task-specific adaptive DP mechanism [16], that adds random noise according to the importance of each feature regarding certain machine learning task.

In this paper, we provide the adaptation of the task-specific adaptive DP mechanism to the different architecture of FL, such as homogeneous and heterogeneous FL. We also provide the comprehensive analysis of the impact of task-specific adaptive DP on the performance of the shared global model under various privacy parameter values and training round numbers.

The rest of the paper is organized as follows. We describe the preliminary information needed to understand FL architecture and task-specific adaptive DP method in section 2. Next, we describe the methodology of the proposed method in section 3. In section 4, we provide the comprehensive evaluation of the experiments, and conclude the paper in section 5.

II. Background & Related Work

2.1 Federated learning (vertical and horizontal)

FL is defined as decentralized training of a shared global model by multiple collaborators while keeping the data at the local devices. To implement FL in practice, there are several architectures related to the data split, such as vertical, horizontal, and federated transfer learning.

2.1.1 Horizontal FL

The term horizontal or homogeneous FL refers to the horizontal split of the data, where the data on different devices share the same feature space, but differ in samples. For instance, let us consider the example provided in Figure 1 (a), where multiple hospitals (collaborators) have data

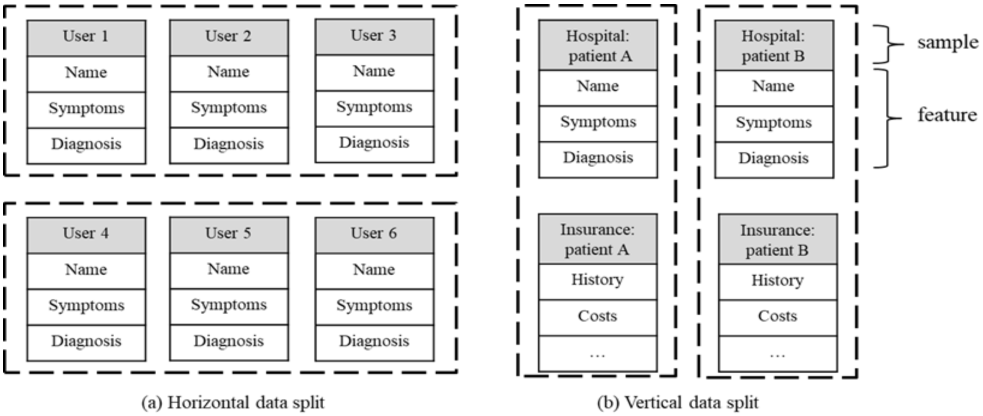


Fig. 1. Example of the (a) horizontal and (b) vertical data partitioning scheme

related to the same set of medical features such as patient name, symptom, and diagnosis, but the individual patient data varies across hospitals due to the different populations.

2.1.2 Vertical FL

The term vertical or heterogeneous FL refers to the vertical split of the data, where the data on different devices share the same samples, but differ in feature space. For instance, let us consider the case of hospital and insurance company collaborating on a FL task.

In vertical FL setting, both hospital and insurance company may have data related to the same set of individuals (patients), but differ in terms of set of feature. For example, hospital's dataset consists of medical data such as patient's name, symptoms, and diagnosis as shown in Figure 1 (b). While the insurance company's dataset consists of the data related to the financial aspect of the same set of patients, such as cost of medical treatments, medical history, and other details.

2.2 Differential Privacy

DP is the mathematical notion of privacy, that bounds the information leakage in neighboring datasets, controlled by the privacy parameter ε and relaxation parameter δ.

Definition 1. (ε, δ - DP) A randomized algorithm K gives ε, δ-DP if for all data sets $D_1$and D_2 differing on at most one element, and all $S \subseteq Range(K)$,

$$\frac{\Pr[K(D_1) \in S]}{\Pr[K(D_2) \in S]} \leq \exp(\varepsilon) + \delta \tag{1}$$

Here, the inequality 1 states that the absence or presence of single record in the dataset can cause at most $\exp(\varepsilon) + \delta$ change in the output.

The basic idea behind DP is to add random noise in order to achieve single record indistinguishability in neighboring datasets. The amount of random noise is determined by sensitivity metric, that is defined as follows:

Definition 2. Sensitivity of a function $f: D \to R$ that maps the dataset D to the real number R is defined as the maximum difference between $f(D_1)$ and $f(D_2)$ as follows:

$$\triangle f = \max \ \| f(D_1) - f(D_2) \| \qquad (2)$$

where $\| \bullet \|$ refers to the norm, i.e l_1 or l_2 norm. Sensitivity metric $\triangle f$ is the maximum change caused by the absence or presence of a single record.

DP mechanism is a practical way to add and control the amount and type of random noise. One of the representative DP mechanisms is called Laplace mechanism, that adds noise drawn from the Laplace random distribution to the true value of the answer. The formal definition is as follows:

$$f'(x) = f(x) + Lap(\frac{\triangle f}{\varepsilon}) \qquad (3)$$

Here, the $f(x)$ is the true answer and $Lap(\frac{\triangle f}{\varepsilon})$ is the random noise calibrated by the sensitivity and privacy parameter values.

Such addition of DP noise is highly effective in case of database queries over the large data. However, in case of addition of noise directly to data, that is the most straightforward way of data protection, it negatively affects the utility of data in further usage. This problem is usually referred to as privacy and utility trade-off problem.

2.2.1 Task-specific Adaptive DP

Task-specific adaptive DP [16] was introduced as a method to mitigate the impact of privacy and utility trade-off problem by adaptively adding noise according to the importance of each feature according to the machine learning task. The key concept is to add less noise to the more important features, and add more noise the less important features. In practice, the noise is calibrated with adaptive sensitivity metric of the element that is defined as follows:

$$\triangle AS_{j_i} = \triangle f_j - \triangle i_i \qquad (4)$$

Here, the $\triangle f_j$ is the traditional sensitivity of the record j, and $\triangle i_i$ is the importance of the feature i estimated by explainable artificial intelligence (XAI) technique.

This method was proven to be effective in regular machine learning setting.

III. Task-specific Adaptive DP in FL setting

In this section, we explain the details of extension of task-specific adaptive DP into the FL setting, analyse the impact of different data split type and data distribution among clients, provide performance analysis, and discuss the challenges.

3.1 Integration of task-specific adaptive DP into FL environment

Let us overview the schema to integrate the task-specific adaptive DP method into FL environment as shown in Figure 2. The noise is added to the local train data of each client during the first round, and calibrated by the adaptive sensitivity proposed in [16]. Here, the record sensitivity is set to 1, since adding or removing a single record would result in change of by at most 1. The feature importance is determined with XAI model explainer such as SHAP applied to each local model, and privacy parameter is agreed by all clients.

The key steps of integration of task-specific adaptive DP in FL setting is shown in Figure 2 and can be described as follows:

1. Initialize and distribute the model M to all clients

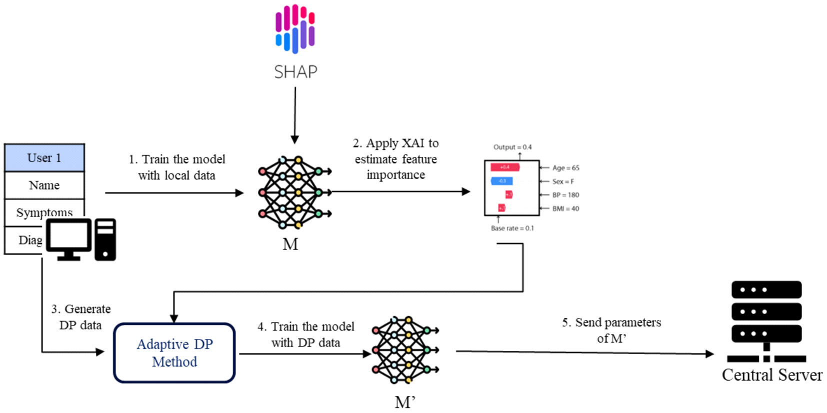


Fig. 2. Task-specific adaptive DP in FL environment

2. Each client trains the model M with original train data
3. Each client estimates the feature importance by applying XAI feature explainer to the trained model M
4. Each client adds task-specific adaptive DP noise to the original train data calibrated by
5. Each client trains another model M’ with identical parameters and sends back to the server
6. Central server aggregates the received models
7. Updated model is sent to all clients
8. Iterate until model converges
9. Output the final aggregated model

From this setting, the differentially private noisy data is generated only once during the first round of the training. However, if new data is added, additional noise is applied to the newly added data. The privacy parameters are summed up according to the sequential composition property of DP.

3.2 Challenges related to data split and distribution

In this section, we discuss the challenges and assumptions associated with difference of the data split type and distribution.

In horizontal FL setting, we assume that each client owns data with the same feature set, but with different sample set.

When the data is independent and identically distributed (i.i.d) among clients, we assume that the importance of each feature is similar for all clients, as shown in Figure 5. Thus, the DP noise is added similarly in each client side. and the model is converged after multiple iterations.

However, when the data is not i.i.d, both the performance of the global model and convergence time are affected a lot. For instance, let us assume, the situation when the data is skewed according to certain feature, leading to non-uniform distribution of the data. Due to such imbalanced data distribution among clients, the feature importance in the first round varies significantly. Hence, the amount of random noise to be added is also different.

In case of vertical data split, we assume that each client owns data with different feature set, but with same sample set. Hence, the feature importance of each client differs in all rounds of the training process.

IV. Experiment

In this section, we provide the experimental evaluation of the task-specific adaptive DP method in FL environment, to answer the following research questions:

- RQ1: How does the task-specific adaptive DP method affect the performance of the global model?
- RQ2: How does different XAI techniques affect the performance of the global model?
- RQ3: How does task-specific adaptive DP compare to the state-of-the-art DP methods in FL setting?
- RQ4: How does the data partitioning scheme (horizontal and vertical) affect the efficiency of task-specific adaptive DP method?

To measure the performance we run experiments on Windows 10, AMD Ryzen 5 3600 6-core processor, 16 Gb RAM, NVIDIA(R) GeForce RTX 2080 Ti GPU, and Python-3.10. To simulate the training process of the FL environment, we utilized the Tensorflow Federated (TFF) library.

RQ1. How does the task-specific adaptive DP method affect the performance of the global model?

To evaluate the practical efficiency of the task-specific adaptive DP in the FL setting, we run several experiments in horizontal data partitioning scheme with i.i.d assumption under varying privacy parameter value ε from 0.1 to 10 and number of clients from 3 to 10.

Specifically, we used the benchmarking dataset Adult Census Income dataset [17], that consists of nearly 48,000 records with 14 numerical features. We trained the artificial neural network with original data and reached the baseline accuracy of 85% in regular machine learning setting.

In case of FL setting, the accuracy of the model trained and tested on original data under 3 client setting was by as much as 80.18%, 82.09%, and 84.30% for 5, 10, and 20 rounds. In case of 5 client setting, the accuracy of the model was by as much as 80.63%, 81.75%, and 84.21% for 5, 10, and 20 rounds. In case of 10 client setting, the accuracy of the model was by as much as 80.37%, 81.67%, and 84.56% for 5, 10, and 20 rounds, respectively.

Next, we used the SHAP [18] model explainer to each local model during the first round. The feature importance scores were then normalized, such that the sum of all scores does not exceed 1.

Differentially private data was generated by adding random noise, sourced from a Laplace distribution, to each element. The noise was calibrated using the privacy parameter ε and adaptive sensitivity values. Table 1 illustrates the performance of the global model under various conditions.

Specifically, with ε = 0.1-differentially private data, the model recorded performance rates of 82.1%, 82.0%, and 83.3% after 5, 10, and 20 rounds of training, respectively, when the number of clients was 3. When the client count increased to 5, the performance metrics were 81.2%, 80.9%, and 82.1% after 5, 10, and 20 rounds of training, respectively. Likewise, with 10 clients, the

Table 1. Performance of the global model with data generated by task-specific adaptive DP method

	N = 3			N = 5			N = 10		
rounds	5	10	20	5	10	20	5	10	20
ε = 0.1	82.1%	82.0%	83.3%	81.2%	80.9%	82.1%	80.8%	81.4%	81.4%
ε = 1	84.9%	85.0%	84.9%	81.7%	81.4%	81.8%	81.9%	83.7%	83.8%
ε = 10	84.6%	85.3%	85.7%	84.8%	85.2%	85.2%	83.8%	85.1%	85.1%

performance rates stood at 80.8%, 81.4%, and 81.4% after 5, 10, and 20 rounds of training.

In the scenario with ε = 1-differentially private data, the model exhibited performance rates of 84.9%, 85.0%, and 84.9% after 5, 10, and 20 rounds of training when the number of clients was 3. In cases where the number of clients was 5, the model reached performance levels of 81.7%, 81.4%, and 81.8% after 5, 10, and 20 rounds, respectively. For a client count of 10, the performance metrics were 81.9%, 83.7%, and 83.8% after 5, 10, and 20 rounds of training.

In the case of ε = 10-differentially private data, the model's performance also varied with the number of training rounds and participating clients. Specifically, with 3 clients, the model yielded performance rates of 84.6%, 85.3%, and 85.7% after 5, 10, and 20 rounds of training, respectively. When the number of clients increased to 5, the performance metrics were 84.8%, 85.2%, and 85.2% after 5, 10, and 20 rounds, respectively. With a client count of 10, the model achieved performance rates of 83.8%, 85.1%, and 85.1% after completing 5, 10, and 20 training rounds.

From these results, we can observe that the performance of the global model is influenced by the multiple factors, including the privacy parameter ε. The best results are shown under ε = 10, that is similar to the baseline accuracy of global model trained on the original data. In the case of lower ε values such 1 and 0.1, there is decrease in the performance, particularly as the number of participating clients increases. This can be explained by the nature of DP, the smaller is the ε, the more noise is added, thus, the stronger are the privacy guarantees. The average performance degradation for all cases does not exceed 5%, thus, we can conclude that the task-specific adaptive DP method in FL setting provides smaller performance degradation.

RQ2. How do different XAI techniques affect the performance of the global model?

In this section, we evaluate the influence of different XAI techniques on the performance of the global model when noise is added using the task-specific adaptive DP method. We used 3 different XAI techniques such as SHAP [18] that utilizes game theory and the Shapley values, LIME [19] that perturbs the input data and observes the change in the model's output, and ELI5 [20] that provides general approach to explain the model.

Figure 3 illustrates the feature importance of three randomly selected attributes: age, education.num, and marital.-status, as determined by various model explainers. The black color represents the values obtained from the SHAP model explainer, the dark grey color corresponds to the values from the LIME model explainer, and the light grey color indicates the values from the ELI5 model explainer. Notably, while the SHAP and ELI5 explainers provide somewhat similar results, the LIME model explainer's results are significantly different.

Such discrepancy has also impacted the

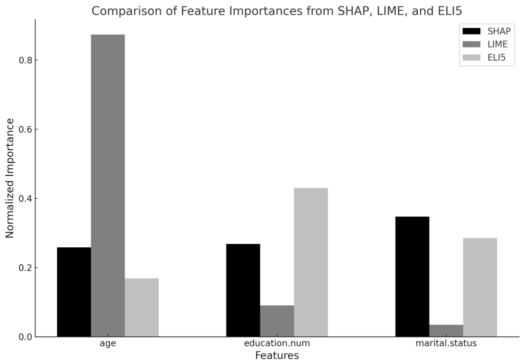


Fig. 3. Comparison of the feature importance estimated by different XAI techniques

amount of DP noise added to the data, resulting in significant changes in the performance of the global model. Table 2 presents the performance of the global model when different XAI tools were utilized to generate DP training data with three clients, and privacy parameter ε were 0.1 and 1. Using the SHAP model explainer, the accuracy of the global model was 82.1% and 84.9% for privacy parameters 0..1 and 1, respectively. With the LIME model explainer, the accuracy was lower, at 63.3% and 64.2% for or ε = 0.1 and 1, respectively. In the case of ELI5 model explainer, the accuracy of the global model was 79.5% and 82.3% for ε = 0.1 and 1, respectively.

It is important to note that the complexity of the task-specific adaptive DP method in the FL setting largely depends on the XAI technique used. From the experimental results, both the SHAP and ELI5 model explainers showed high computational demands, while using LIME was significantly more computationally efficient. These experimental results make it evident that choosing appropriate XAI tools is crucial for generating high-quality data with a task-specific adaptive DP method. In subsequent experiments, we utilized the SHAP model explainer, as its game-theoretic approach provides consistent and reliable estimations of feature importance, even if it requires more computational resources.

Table 2. Performance of the global model with different XAI tools used to generate DP data

XAI tool	ε = 0.1	ε = 1
SHAP [18]	82.1%	84.9%
LIME [19]	63.3%	64.2%
ELI5 [20]	79.5%	82.3%

RQ3. How Does Task-Specific Adaptive DP Compare to Traditional Laplace DP in FL setting?

In this section we compare the task-specific adaptive DP mechanism in the FL setting with other DP methods such as traditional Laplace DP mechanism and DPGAN [21] in the FL setting. Specifically, we compare the performance of the global models trained on the data generated by both of these methods under various ε values.

Figure 4 represents a comparative analysis of accuracy and binary cross-entropy loss across training rounds in FL. Models trained on data generated via the traditional Laplace mechanism are represented by black dashed lines, data generated via the DPGAN method are represented by blue dashed lines, while those using the task-specific adaptive DP mechanism are indicated by solid green lines.

As we can observe from top row of Figure 4, the accuracy of the model trained on data generated with task-specific adaptive DP method is noticeably higher compared to the accuracy of the models trained on data generated with traditional Laplace DP mechanism, and those generated by DPGAN method. The average difference in the accuracy metric for ε = 0.1 was by as much as 20.2% and 11.5%, for ε = 1 was by as much as 15.6% and 9.8%, and for ε = 10 was by as much as 9.6% and 5.6% for train data generated by Laplace DP and DPGAN methods, respectively.

Whereas from the bottom row of the Figure 4, we can observe that the binary cross-entropy loss for the model trained with data generated by traditional Laplace DP mechanism and DPGAN method are significantly higher compared to the loss of the model trained on data generated with task-specific adaptive DP mechanism. The average difference in the binary cross-en-

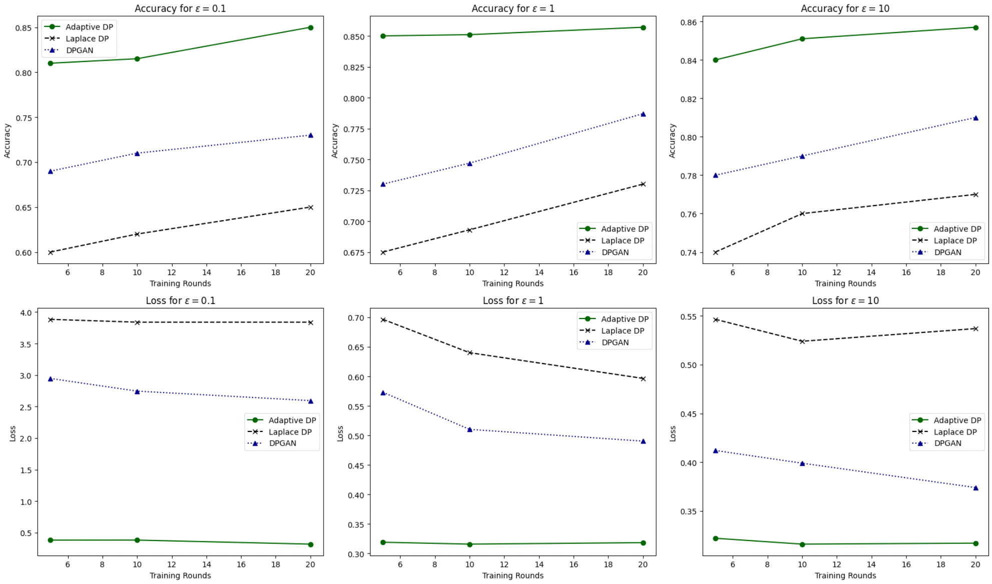


Fig. 4. Comparison of the accuracy and loss across training rounds for various DP data generation methods

tropy loss metric for ε = 0.1 was by as much as 3.495 and 2.403, for ε = 1 was by as much as 0.327 and 0.207, and for ε = 10 was by as much as 0.217 and 0.077 for train data generated by Laplace DP and DPGAN methods, respectively.

From these results, it is evident that the task-specific adaptive DP method outperforms both the traditional Laplace DP mechanism and DPGAN method, particularly at smaller ε values such as 0.1. This suggests that the adaptive method offers a compelling balance of high performance and robust privacy guarantees in the FL environment.

RQ4. How does the data partitioning scheme (horizontal and vertical) affect the efficiency of task-specific adaptive DP method?

In this section we explore the specifics of the data partitioning scheme and its influence on the performance of task-specific adaptive DP by comparing the feature importance and global model performance for horizontal and vertical split.

Horizontal data partitioning scheme. Figure 5 displays the feature importance of four randomly selected attributes: age, education.num, marital.status, and hours. per. week, in a horizontal FL setting under the i.i.d assumption. Subfigure (a) illustrates the feature importance after the initial training round, while subfigure (b) presents the feature importance after the fifth training round. From these subfigures, it is evident that the feature importance remains relatively stable across multiple rounds of training. Moreover, the importance attributed to each feature is consistent across different clients.

Figure 6 displays the feature importance of four attributes in a horizontal FL setting under the non i.i.d assumption. Here, data is skewed based on the ′occupation′ attribute. Specifically, Client 1 possesses data primarily related to certain occupations, while Client 2 and Client 3 have data corresponding to another set of occupations. Subfigure (a) depicts the feature im-

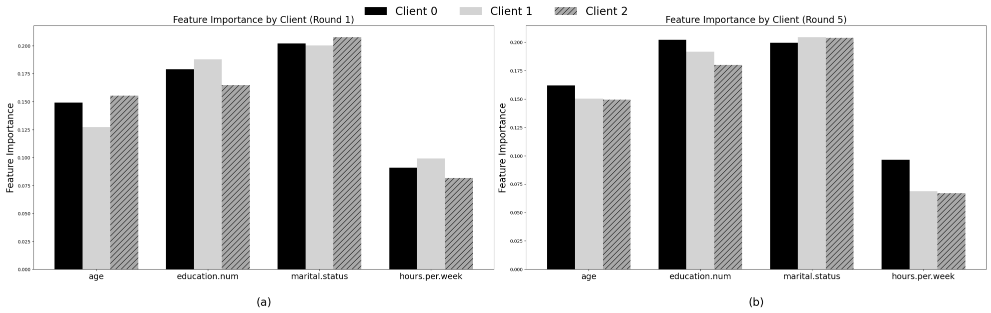


Fig. 5. Feature importance in data i.i.d assumption for (a) after the first round and (b) after the fifth round

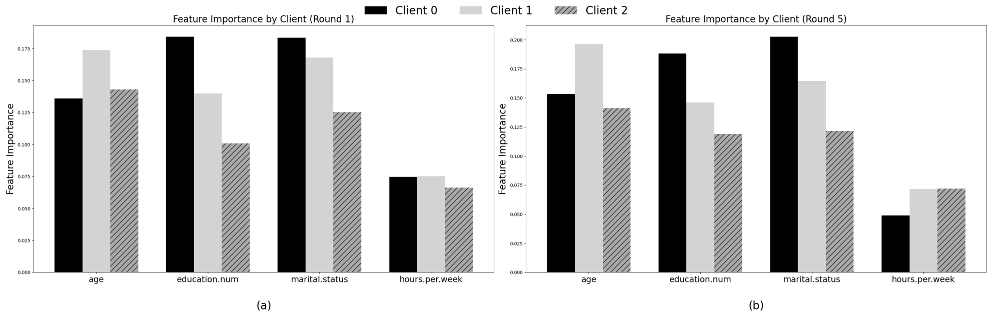


Fig. 6. Feature importance in data non i.i.d assumption for (a) after the first round and (b) after the fifth round

portances after the first round, while subfigure (b) represents the feature importances after the fifth round. From these subfigures, it is evident that the feature importance for each client differs significantly and also undergoes changes after 5 rounds.

The baseline accuracy of the global model trained on the original data with non i.i.d assumption was 81.5%, 82.2%, and 82.8% for rounds 5, 10, and 20 when number of clients was 3. We can observe that performance of the global model with non i.i.d assumption is lower compared to the performance of the global model with i.i.d assumption.

When applying task-specific adaptive DP noise to the train data, we observed the following results for ε = 0.1 86.3%, 86.8%, and 88.7% after rounds 5, 10, and 20. For ε = 1, the performance of the global model was 87.54%, 90.68%, and 91.05% after rounds 5, 10, and 20, respectively. In case of the ε = 10, the performance of the global model was 89.7%, 90.0%, and 90.6% after rounds 5, 10, and 20, respectively.

In this study, we also consider the case, where clients may have different data distributions, such as 2 clients hold i.i.d data, while other hold non i.i.d data. The baseline accuracy of the global model trained on the original data with mixed data distribution assumption was 72.3%, 73.4%, and 74.7% for rounds 5, 10, and 20 with 3 clients. Notably, the model struggles to learn the mixed data distribution, leading to significantly lower performance.

When applying task-specific adaptive DP noise to the train data with mixed data distribution, we observed the following results for ε = 0.1 75.7%, 76.0%, and 77.1% for rounds 5, 10, and 20. For ε = 1 the accuracy of the global model was 77.3%, 77.8%, and 78.5% for rounds 5, 10, and 20, respectively. For ε = 10, the performance of the global model was 81.1%, 82.4%, and 82.9% after rounds 5,10, and 20.

Table 3 summarizes the accuracy metrics for different data distribution cases when the number of clients was 3, and the number of training rounds was 20. From these results, we can observe that the application of task-specific adaptive noise, which emphasizes the importance of the feature to the global model, increased the performance of the global model compared to the baseline results in both non i.i.d and mixed data distribution cases.

Table 3. Performance of the task-specific adaptive DP method with horizontally partitioned data with different data distribution assumptions after round 20

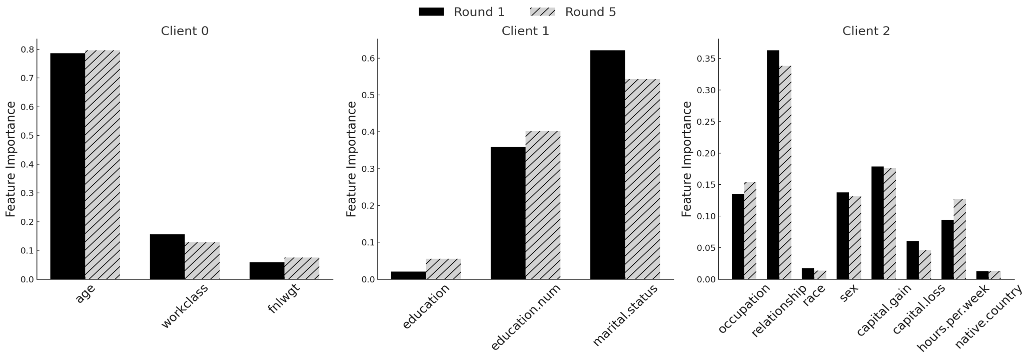
	i.i.d	non i.i.d	mixed
baseline	84.3%	82.8%	74.7%
ε = 0.1	83.3%	88.7%	77.1%
ε = 1	84.9%	91.0%	78.5%
ε = 10	85.7%	90.6%	82.9%

Vertical data partitioning scheme. In the case of vertical data partitioning, we assume that each client owns a distinct set of features for the same samples. For example, in a scenario involving 3 clients, client 0 and client 1 each possess data comprising three attributes, while client 2 owns the remaining features.

Consequently, we do not evaluate feature importance on a per-client basis in this paper, nor do we consider assumptions about data distribution. Instead, we focus on comparing feature importances after the first and fifth rounds, as depicted in Figure 7. Notably, while the feature importance for client 0 remains relatively stable throughout the training, the importance of features for client 3 experiences tangible changes after five rounds.

The baseline accuracy of the global model, when trained on the original data, stood at 79.2%, 79.2%, and 80.1% after 5, 10, and 20 rounds, respectively.

Upon applying task-specific adaptive DP noise to the training data, we observed the following results: Under ε = 0.1, the accuracy was 74.7%, 74.9%, and 75.1% after 5, 10, and 20 rounds. For ε = 1, the accuracy rates were 76.6%, 77.0%, and 77.5% after 5, 10, and 20 rounds. In the case of ε =10, the performance was 79.0%, 79.2%, and 79.3% after 5, 10, and 20 rounds.



(a) (b) (c)

Fig. 7. Feature importance for (a) client 0, (b) client 1, and (c) client 3

From these experimental results, it can be observed that the performance of the task-specific adaptive DP method in a vertical FL environment is similar to that in a horizontal FL environment.

V. Conclusion

In this study, we have rigorously evaluated the application of a task-specific adaptive DP mechanism within the FL environment. Through comprehensive performance analysis backed by experimental results, we demonstrated several important findings. First, the task-specific adaptive DP outperforms the straightforward Laplace DP mechanism and data generated by DPGAN model, underscoring its effectiveness in enhancing privacy without significantly compromising model performance.

Second, while the adaptive DP shows results that are slightly lower than the baseline accuracy in the context of horizontal FL with i.i.d data assumptions and vertical FL, these differences were minimal, confirming the utility of this approach in various FL architectures.

Most intriguingly, we observed that the performance of models trained with task-specific adaptive DP in horizontal FL with non i.i.d and mixed data assumptions was even superior to the baseline models. This suggests that emphasizing feature importance in a non-i.i.d data distribution can be advantageous, offering a new avenue for future research in optimizing the balance between privacy and utility in FL.

These findings not only validate the use of task-specific adaptive DP as a viable privacy-preserving mechanism in FL, but also provide a background for the practical deployment of this mechanism in various industrial contexts. In sectors like healthcare and finance that routinely handle sensitive information and different data and feature samples, this approach is especially pertinent.

References

〔1〕 S. R. Pandey, N. H. Tran, M. Bennis, Y. K. Tun, A. Manzoor and C. S. Hong, "A Crowdsourcing Framework for On-Device Federated Learning," in IEEE Transactions on Wireless Communications, vol. 19, no. 5, pp. 3241-3256, May 2020,

〔2〕 J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang and M. Guizani, "Reliable Federated Learning for Mobile Networks," in IEEE Wireless Communications, vol. 27, no. 2, pp. 72-80, April 2020

〔3〕 J. Li et al., "A Federated Learning Based Privacy-Preserving Smart Healthcare System," in IEEE Transactions on Industrial Informatics, vol. 18, no. 3, pp. 2021-2031, March 2022,

〔4〕 M. Ali, F. Naeem, M. Tariq and G. Kaddoum, "Federated Learning for Privacy Preservation in Smart Healthcare Systems: A Comprehensive Survey," in IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 2, pp. 778-789, Feb. 2023

〔5〕 G. Shingi, "A federated learning based approach for loan defaults prediction," 2020 International Conference on Data Mining Workshops (ICDMW), pp. 362-368, Nov. 2020

〔6〕 Ahmed Imteaj, M. Hadi Amini, "Leveraging asynchronous federated learning to predict customers financial distress", Intelligent Systems with Applications, Volume 14, pp. 20064, May 2022,

〔7〕 Briland Hitaj, Giuseppe Ateniese, and

Fernando Perez-Cruz. “Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning.” In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS ′17), pp. 603–618, Oct. 2017

〔8〕 Zhu, Ligeng, Zhijian Liu, and Song Han. ″Deep leakage from gradients.″ Proceedings of Neural Information Processing Systems (NIPS'19), pp. 14774–14784, Dec. 2017

〔9〕 Tolpegin, Vale, et al. ″Data poisoning attacks against federated learning systems.″ In proceedings of 25th European Symposium on Research in Computer Security, ESORICS 2020, pp. 480-501, Sep. 2020

〔10〕 V. Shejwalkar, A. Houmansadr, P. Kairouz and D. Ramage, ″Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning,″ In proceedings of 2022 IEEE Symposium on Security and Privacy (SP), pp. 1354-1371, May 2022

〔11〕 L. Shi, J. Shu, W. Zhang and Y. Liu, ″HFL-DP: Hierarchical Federated Learning with Differential Privacy,″ In proceedings of 2021 IEEE Global Communications Conference (GLOBECOM), pp. 1-7, Dec. 2021

〔12〕 K. Wei et al., ″Federated Learning With Differential Privacy: Algorithms and Performance Analysis,″ in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3454-3469, Apr. 2020

〔13〕 Park June Beom, & Park Jong Sou. “An Implementation of Federated Learning based on Blockchain”. The Korean Journal of BigData, vol. 5(1), pp. 89-96. Aug. 2020

〔14〕 R. Kanagavelu et al., ″Two-Phase Multi-Party Computation Enabled Privacy-Preserving Federated Learning,″ In proceedings of 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), pp. 410-419, May 2020

〔15〕 Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu and X. Zheng, ″Privacy-Preserving Federated Learning Framework Based on Chained Secure Multiparty Computing,″ in IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6178-6186, Apr. 2021

〔16〕 Utaliyeva A, Shin J, Choi YH. “Task-Specific Adaptive Differential Privacy Method for Structured Data”, in Sensors MDPI, vol.23 no.4, pp.1980-1992, Feb.2023

〔17〕 Becker, Barry and Kohavi,Ronny. (1996). Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20.

〔18〕 Scott M. Lundberg and Su-In Lee. 2017. “A Unified Approach to Interpreting Model Predictions”. In proceedings of Neural Information Processing Systems (NIPS'17), pp. 4766-4777, Dec. 2017

〔19〕 Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. “Why should i trust you? Explaining the predictions of any classifier.″ in Proceedings of the 22nd ACM SIGKDD, pp.1135-1144, Aug. 2016

〔20〕 ELI5: A Library for Debugging/Inspecting Machine Learning Classifiers 〔Software〕. Available from https://github.com/TeamHG-Memex/eli5. Accessed on 18th Jan. 2024

〔21〕 Xie, L., Lin, K., Wang, S., Wang, F., and Zhou, J. “Differentially private generative adversarial network“ arXiv:1802.06739, Feb. 2018

〈저자소개〉

우타리예바 아쎔 (Assem Utaliyeva) 학생회원
2020년 2월: 부산대학교 정보컴퓨터공학부 학사
2022년 2월: 부산대학교 정보융합공학과 석사
2022년 3월~현재: 부산대학교 정보융합공학과 박사과정
〈관심분야〉 AI 보안, 정보보호, 차분 프라이버시

최 윤 호 (Yoon-Ho Choi) 종신회원
2008년: 서울대학교 전기컴퓨터공학부 박사
2010년: 펜실베니아 주립대학교 박사후 연구원
2012년: 삼성전자 네트워크사업부 책임연구원
2014년: 경기대학교 융합보안학과 조교수
2016년~현재: 부산대학교 정보융합공학과 교수
〈관심분야〉 지능형 악성코드분석, 코드보안, 소프트웨어 취약성분석, 프라이버시, AI 보안, 블록체인, 유무선 네트워크 보안 등