# The Price of Forgetting: Data Redemption Mechanism Design for Machine Unlearning

Yue Cui
Department of Computer Science
City University of Hong Kong
Hong Kong, China
yuecui8-c@my.cityu.edu.hk

Man Hon Cheung
Department of Computer Science
City University of Hong Kong
Hong Kong, China
mhcheung@cityu.edu.hk

## ABSTRACT

Massive data is gathered together in the server for the training of machine learning models, which brings potential privacy risks for users. Recent regulations, such as GDPR, require the server to provide *the right to be forgotten* for users. Existing studies about machine unlearning can technically achieve such a requirement but inevitably incurs a loss on the server side from both the time spent on performing machine unlearning and accuracy loss on the model. Without sufficient compensation for the server, it has no incentive to perform machine unlearning effectively and honestly. In this paper, we propose the first incentive mechanism for data redemption that maximizes the profits for the server and users to achieve a win-win combination to the best of our knowledge. Specifically, we formulate a two-stage decision process, where the server optimizes the unit price function of the compensation for its profit maximization. Each user then decides his data redemption amount based on the server's announced price to improve its utility-payment trade-off. We formulate both time and accuracy cost functions with real data. The analysis is challenging as the individual user's data redemption decision together affects the server's pricing decision. Also, the decision problems for both the server and users are non-convex. Nevertheless, we are able to derive the closed-form solutions in both stages through characterizing the convexity of the server's profit function and users' payoff function and dividing them into sub-problems. The experimental results on two real-world datasets demonstrate that our solutions can achieve a win-win situation and the major data redemption cost happens in the accuracy loss.

## 1 INTRODUCTION

### 1.1 Motivations

Businesses such as Facebook and Google utilize the data gathered from massive users to train sophisticated models and benefit our life these days. These applications are ubiquitously deployed in our surroundings, such as natural-language understanding [21], shopping recommendation [17], smart healthcare [12], and smart city [24]. However, the well-trained models potentially remember the information of users, including their medical records [22] and personal emails [9]. The potential privacy risks cannot be ignored, therefore, data should be able to be deleted if needed. Fortunately, there are recent data protection regulations, such as General Data Protection Regulation (GDPR) [1] in European Union, California Consumer Privacy Act (CCPA) [2] in the USA, and PIPEDA privacy legislation [3] in Canada, enforcing *the right to be forgotten* and formulating related regulations. These regulations require the models can *unlearn* the learned information from data. It means the models should be able to adapt to a new model that achieves equivalent (or comparable) performance on its task but trained on the dataset that excludes the targeted data. Such a process is known as *machine unlearning* [5, 7, 23].

Machine unlearning can be prohibitive in terms of the *consumed time*, especially for large models and frequent requests [16]. Given a trained model, fully retraining models after removing the target data from the training set is the most legitimate way, but it leads to a large time overhead. For example, finishing a 90-epoch ImageNet-1k training with ResNet-50 on a P100 GPU takes 10 days [10]. SISA [5] proposed by Bourtoule *et al.* speeds up the machine unlearning by up to 4 times on the Purchase dataset and 2 times on the SVHN dataset by sharding and slicing the dataset. However, the time spent on performing machine unlearning is still significant. The cost of time causes a large financial loss for the companies which prevents the companies from performing machine unlearning consciously after receiving requests.

Another key factor that makes companies hesitate to perform machine unlearning is the *loss of model accuracy*. As mentioned in [5], machine unlearning is challenging because the training is stochastic and we have a limited understanding of how each data point impacts the model. It is not obvious how to modify the model parameters, especially for complex models. Such modification inevitably affects other data and thus degrades the model's accuracy. Experiments conducted by Ghorbani *et al.* [14] on the Breast Cancer dataset show that removing 10% of training data can result in around 15% prediction accuracy, which is a significant drop. The recent study SISA [5] induces 16.14% of the average top-5 accuracy on Imagenet with ResNet-50. The accuracy drop of the model resulting from removal requests severely affects the performance of the trained model, which makes the companies extremely unwilling to perform machine unlearning.

However, existing regulations require the companies to perform machine unlearning once the data removal requests are received without any compensation. Without the consideration of an *economic mechanism* to make sufficient reimbursement to the companies, they have no incentive to perform machine unlearning consciously given the above-mentioned time and accuracy costs. The challenges in designing an effective economic mechanism to compensate the server considering the cost of performing machine unlearning are still an open problem. On the users' side, given the compensation to spend, how much data the users should request to be removed in order to achieve the minimum privacy leakage risks while spending less money is also a challenging problem.

### 1.2 Contributions

To address the above questions, to the best of our knowledge, we propose the first incentive mechanism to compensate for the server's

time and accuracy costs of the server to perform data removal in machine unlearning in this paper. We consider a data redemption mechanism with a server (i.e., data collector and machine unlearning executer) and multiple users (i.e., data providers and machine unlearning requesters). To design a fair incentive mechanism for machine unlearning, one important question is *how much should the server charge the users for the data removal?* If the price is low, the costs of time and accuracy cannot be sufficiently compensated. On the other hand, if the price is too high, the users are not willing to participate anymore and the server has to forgo the chance of earning profitable compensation. More specifically, for profit maximization, *how should the server decide the price to tradeoff between the compensation earned from the users and the data removal cost due to the time cost and accuracy loss?*

Combining the above issues, on the server side, we take three components into its profit function: 1) the sum of the received compensations from users, 2) the time cost of performing machine unlearning, and 3) the accuracy cost after performing machine unlearning. In this paper, the time cost and accuracy cost functions of machine unlearning are modeled by fitting the results from real-world dataset (i.e., MNIST [13] and Cifar-10 [20]). The unit price of each data to be removed is modeled as a linear decreasing function for the server to optimize to maximize its profit. On the users' side, two components are considered in their profit function: 1) the utility gained for a reduced privacy leakage risk by removing the data, and 2) the compensation to be paid to the server. We formulate the utility function of removed data following existing studies. The data redemption amount is optimized by the users in order to maximize their payoffs.

We model the data redemption mechanism of the server and users as a two-stage decision process, where Stage I is the server's profit maximization problem by optimizing the unit price function parameter and Stage II is the users' payoff maximization problem by optimizing the data redemption amount. Since the server and users are making decisions individually, we solve the decision problem of users before the server's and then induce the solution of Stage II backward to Stage I's decision problem. Solving the induced maximization problem is highly non-trivial since the objective function of Stage I is non-convex after substituting Stage II's solution to it. We decompose the problem into two sub-problems on different ranges of the unit price function parameter. Specifically, we characterize the first sub-problem as a concave optimization problem while the second sub-problem is a linear decreasing maximization problem of the unit price parameter. Thus, we are able to obtain closed-form by comparing the profits under these two sub-problems.

Despite the complexity of the independent decision-making process, we derive the optimal solutions for both the server and users in closed form and demonstrate some interesting insights on the optimal pricing mechanism. We summarize our major contributions as follows:

- *Novel mechanism for data redemption*: To the best of our knowledge, this is the first incentive mechanism for machine unlearning, which captures two major costs of machine unlearning (i.e., time and accuracy costs).
- *Independent decision processes*: We formulate the decisions of the server and users as two independent problems, where the server is maximizing its profit and users are maximizing their payoffs.
- *Characterization of time and accuracy cost functions based on real data*: We model the time and accuracy cost function with real-world datasets through experiments.
- *Closed-form solutions for both the server and users*: To solve the non-convex Stage I, we divide it into two sub-problems. We characterize the concavity and linearity of the first and second sub-problem separately to derive closed-form solutions.
- *Insights of optimal pricing mechanism*: Our experiments show that our proposed optimal mechanism achieves higher server's profit and users' payoffs than three baselines. Also, we find that the accuracy cost accounts for the majority of the machine unlearning cost if the server considers the two costs equally important.

## 1.3 Related Works

*1.3.1 Machine Unlearning.* The removal of specific training samples from machine learning models has been studied in prior work. Decremental learning [19, 26] was proposed to adopt the traditional models quickly after some data are removed from the training set such as logistic regression (LR), Support Vector Machine (SVM). Tsui *et al.* [26] investigated primal or dual formulation and chose optimization methods for the decremental learning of linear classifiers (e.g., LR and linear SVM). Machine Unlearning [5, 15, 16] has attracted attention recently, which enables the recovery of machine learning/deep learning models from the partially deleted dataset. The most legitimate way to implement machine unlearning is to retrain the whole model from scratch [8]. SISA [5] worked in an ensemble style, which partitions the dataset into several disjoint parts for the training of sub-models and only retrains the sub-model that contains the targeted data. These methods achieve the success of technically addressing the data removal problem. However, the time and accuracy cost of machine unlearning cannot be fully eliminated. Without the consideration of an economic mechanism to motivate the server by sufficient compensation, the server is not willing to perform the machine unlearning. Instead of considering machine unlearning in the technical aspects, we focus on the economic design of the incentive mechanism data redemption.

*1.3.2 Machine Unlearning Costs.* As previously mentioned, there are two types of costs the server would suffer from: 1) time cost to perform machine unlearning and 2) accuracy cost because of the data removal. Existing studies focus on improving one of the costs and fail to consider them together, thus resulting in a bad time-accuracy trade-off. In the trend of machine unlearning, most of them try to minimize the time cost (e.g., the computational cost in centralized machine unlearning, and communication cost in Federated Unlearning). Ginart *et al.* [15] formulated provably efficient deletion algorithms for k-means clustering and achieved significant speed-up. Liu *et al.* [23] proposed an algorithm to remove the impact of any specific data sample from clients in federated learning, requiring storing the intermediate model updates in the process of model training in order to improve the unlearning efficiency. On the other side, Xue *et al.* [27] measures the impact of each client to achieve an accurate model adaption without the consideration of

time efficiency. Although these papers provide solutions to forget samples and remove their impact, they always trade one of the costs to improve another. In our work, we consider both of the costs in our profit function and thus make sure the compensation is sufficient to motivate the server.

The rest of the paper is organized as follows. We describe our two-stage system model in Section 2 and derive the solutions in closed-form in Section 3. We show our parameter investigation and experimental results in Section 4. Finally, we conclude the paper in Section 5.

## 2 SYSTEM MODEL

In this section, we first discuss the notions and formulations of users' data redemption in Section 2.1. Then we model the data removal cost of the server due to the spent time and sacrificed accuracy in Section 2.3. Next, we discuss the decisions of the server and users in two stages in Section 2.4 and Section 2.5, respectively. In Stage I, the server optimizes the parameter of the unit price function for its profit maximization. In Stage II, the users decide the data redemption amount to remove by taking into account their utility.

### 2.1 Users' Data Redemption and Utility Function

As shown in Figure 3, we consider a data redemption mechanism with a server (i.e., data collector) and multiple users (i.e., data providers) who asking for removing their data from the server. Let $\mathcal{I} = \{1, ..., I\}$ be the set of users and let $d_i$ be the amount of data that user $i$ sold to the server. In the data redemption mechanism under machine unlearning, user $i$ requests that an amount of data $x_i \in [0, d_i]$[1] to be removed.

At the same time, the user can gain a certain level of utility according to the amount of data to be removed because of the improved security level. Let $U_i(x_i)$ be the *utility function* of user $i$ if $x_i$ amount of data is removed from the server. $U_i(x_i)$ is expected to be a concave increasing function in $x_i$ with $U_i(0) = 0$. The utility function is defined as follows, which is a widely adopted choice of utility function [18, 25].

*Definition 1 (Utility Function): The utility of users under the amount of removed data $x_i$ is*

$$U_i(x_i) \triangleq -{x_i}^2 + 2d_i x_i. \tag{1}$$

### 2.2 Server's Data Redemption Pricing

On the other side, the server charges each of the removed data units a price for the compensation of the time spent on performing machine unlearning and the loss of model accuracy. Let $P(x_i)$ be the unit price for removing $x_i$ unit of data such that user $i$'s payment to the server is $x_i P(x_i)$. For fairness with all users, we assume that the server announces only one unit price function for all users instead of the heterogenous prices. The unit price function is expected to be a positive decreasing function because the price should never be negative and the server can attract users to participate by setting a lower unit price when their requested amounts are large. We define the unit price function as a linear function[2] for simplicity of the optimization, which is also widely used in literature [11]. The unit price function is defined as follows.

*Definition 2 (Unit Price Function): The unit price under the amount of removed data $x_i$ is*

$$P(x_i) \triangleq k \cdot (D - x_i), D > max\{d_1, ..., d_I\}, k \geq 0, \tag{2}$$

where the constraint on $D$ and $k$ ensures that $P(x_i) \geq 0$.

### 2.3 Characterization of Server's Data Removal Cost based on Real Data

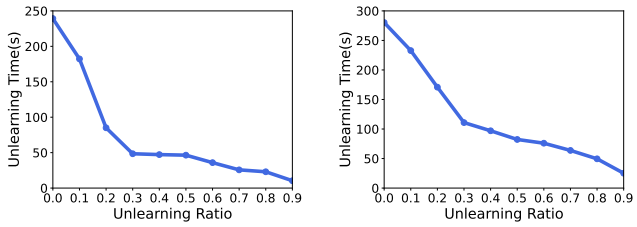
We consider two factors of the server cost resulting from the data removal. Firstly, we consider the time spent on rebuilding the model using the machine unlearning techniques mentioned in Section 1.3.1. Secondly, we consider the accuracy loss of the model resulting from the removed data. The time cost and accuracy cost inevitably exist in any specific machine unlearning methods. We introduce the definition of our time and accuracy cost in the following sections 2.3.1 and section 2.3.2 separately.

*2.3.1 Time Cost.* To investigate the time cost of machine unlearning, we conduct experiments on two popular datasets (i.e., MNIST [13] and Cifar-10 [20]) and use the most legitimate method of machine unlearning (i.e., retrain from scratch [8]). We first train two models on two datasets separately to perform the unlearning. For both of the datasets, we use the model structure with two Convolutional layers and two Fully Connected layers. We train the model for five epochs for the MNIST dataset and 20 epochs for the Cifar-10 dataset to get a satisfactory performance. We range the removed data ratio from zero to 90 percent of the whole training set with a step size of 10 percent and compare the corresponding time spent on retraining the model. Noted that when the removed data is zero, there is no machine unlearning to be performed and the unlearning time is zero. The spent time on performing machine unlearning against the various ratios of removed data are shown in Figure 1. We can observe that the time cost is a convex decreasing function regardless of the type of dataset. This is because the more data is removed, the rest of the data is decreased, thus the retraining time required to spend on such a smaller training set would be less.

We assume that the unlearning algorithm only performs once after collecting all the removal requests from all users because performing multiple times wastes more time. Thus, the time cost will be decided by the total unit of data to remove. We expect the time cost function $T(\sum_{i\in\mathcal{I}} x_i)$ to be a convex decreasing function in normal situations while $T(\sum_{i\in\mathcal{I}} x_i) = 0$ when $x_i = 0, \forall i \in \mathcal{I}$ and $T(\sum_{i\in\mathcal{I}} x_i) = 0$ when $x_i = d_i, \forall i \in \mathcal{I}$. Because the time spent on performing machine unlearning will be zero if no data is removed or all data is removed. But in other situations, the more data to be removed, the less time machine unlearning requires as shown in Figure 1. We use the quadratic function which can provide a good fitness to the actual time cost. We define the time cost function as follows.

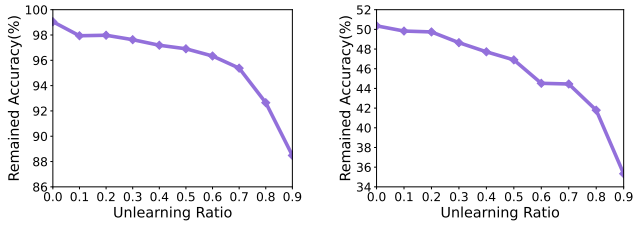
[1] The removed data cannot exceed the sold data or be negative. Noted that $x_i$ does not require to be an integer because it denotes the units of data, such as the MB.

[2] Other forms of unit price function are interesting to explore in future work.




**(a) MNIST** **(b) Cifar-10**

**Figure 1: Unlearning time on two datasets.**




**(a) MNIST** **(b) Cifar-10**

**Figure 2: Remained accuracy of the models on two datasets.**

*Definition 3 (Time cost function): The time cost of the server spent on unlearning under the sum of all $x_i$ is*

$$T(s) \triangleq \begin{cases} 0, & x_i = 0, \forall i \in \mathcal{I} \\ t \cdot (s - \sum_{i \in \mathcal{I}} d_i)^2, & otherwise \end{cases} \tag{3}$$

*where $s = \sum_{i \in \mathcal{I}} x_i$ and $-\sum_{i \in \mathcal{I}} d_i$ ensures $T(\sum_{i \in \mathcal{I}} x_i) = 0$ when $x_i = d_i, \forall i \in \mathcal{I}$.*

*2.3.2 Accuracy Cost.* Similarly, we also investigate the accuracy loss caused by data removal on MNIST [13] and Cifar-10 [20] datasets after performing machine unlearning (i.e., retrain from scratch [8]). We also range the removed data from zero to 90 percent of the whole training set with a step size of 10 percent. We record the remained accuracy[3] of the model after being retrained by the remaining dataset. The results are shown in Figure 2. We can notice that the accuracy of the retrained model is concavely decreasing as we increase the removed data ratio. This is expected because the more data is removed from the dataset, the fewer data can the model learn from, which results in lower accuracy. The accuracy loss can be considered as the remained accuracy subtracted from the original accuracy, which would be convex increasing as we increase the removed data ratio. Also, we expect the accuracy cost function $A(\sum_{i \in \mathcal{I}} x_i)$ satisfies $A(\sum_{i \in \mathcal{I}} x_i) = 0$ when $x_i = 0, \forall i \in \mathcal{I}$ because there will be zero accuracy loss if no data is removed.

Similar to the above-mentioned time cost, the accuracy cost is also only associated with the total unit of data to remove. Thus, we define the accuracy cost as follows.

[3]The ratio of correctly predicted samples to the total samples.

*Definition 4 (Accuracy cost function): The accuracy cost of the server resulted from unlearning $\sum_{i \in \mathcal{I}} x_i$ unit of data is*

$$A(s) \triangleq a \cdot s^2, \tag{4}$$

*where $s = \sum_{i \in \mathcal{I}} x_i$.*

## 2.4 Stage I: Server's Profit Maximization

The data removal cost is the summation of the time cost and accuracy cost, which can be defined as:

$$C(s) \triangleq T(s) + \theta A(s), \theta > 0, \tag{5}$$

where $\theta$ is a scaling factor for the importance of time cost and accuracy cost. In Stage I, the server optimizes the unit price parameter $k$ to maximize its profit (i.e., received payment minus data removal cost):

$$\begin{aligned} \underset{k}{\text{maximize}} \quad & \pi(k) \triangleq \textstyle\sum_{i \in \mathcal{I}} x_i(k) P(x_i(k)) - C(\sum_{i \in \mathcal{I}} x_i(k)) \\ & = \textstyle\sum_{i \in \mathcal{I}} k \cdot x_i(k)(D - x_i(k)) - C(\sum_{i \in \mathcal{I}} x_i(k)), \\ \text{subject to} \quad & k \leq 0, \end{aligned} \tag{6}$$

where $x_i(k)$ is the response (i.e., data redemption amount) of user $i$ to the unit price function parameter $k$, which will be discussed later in Section 2.5. Here $x_i(k)P(x_i(k))$ is the payment received from user $i$.

## 2.5 Stage II: Users' Payoff Maximization

After receiving the unit price function parameter in Stage I, each user $i$ decides the data redemption amount independently in Stage II to maximize its payoff, which consists of two parts:

- *Utility Gain.* User $i$ gains a utility $U_i(x_i)$ if $x_i$ unit of data is successfully removed.
- *Payment.* User $i$ has to make a payment $x_i P(x_i)$ to the server to compensate for its data removal cost under the decision of the server's unit price function $P(x_i)$.

Overall, the user $i$'s payoff function to be maximized is

$$\begin{aligned} \underset{x_i}{\text{maximize}} \quad & U_i(x_i) - x_i P(x_i). \\ \text{subject to} \quad & 0 \leq x_i \leq d_i. \end{aligned} \tag{7}$$

Noted that there are no direct interactions among users because each user optimizes his amount solely given the unit price function decided by the server. However, their total requested data redemption amount $\sum_{i \in \mathcal{I}} x_i$ collectively affects the server's data removal cost in Equation (5). As the server has the complete knowledge of $d_i, \forall i \in \mathcal{I}$ to optimize the unit price function, there is no chance for any user to hide its real $d_i$ to the server to get a higher payoff.

# 3 SOLVING TWO-STAGE DECISIONS OPTIMIZATION

The process of the data redemption mechanism is shown in Figure 3. Firstly, the server computes its unit price setting $k$ using the knowledge of the previously obtained data (e.g., the mean amount of data $\mu$). Then, the server announces $k$ to all users. Thirdly, all users decide the data redemption amount $x_i$ independently by themselves. Finally, the users request the server to remove $x_i$ unit of their data. In this section, we analyze the optimization as a two-stage Stackelberg game by using backward induction. Specifically, we

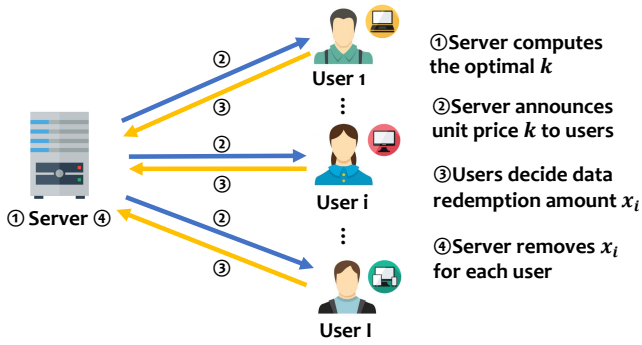



**Figure 3: Illustration of data redemption mechanism.**

first analyze the optimal solution of Stage II in Section 3.1 and then plug in the solution of Stage II into Stage I to get the solution of Stage I in Section 3.2.

## 3.1 Solutions of Stage II

In this section, we provide the solution of $x_i$ for each of the users when the unit price function parameter $k$ is given by the server. We first analyze the convexity of Stage II and then solve the problem separately.

We utilize secondary conditions to analyze the convexity of the objective function of Equation (7) and derive the conditions of its convexity as *1) when* $0 \le k < 1$, *the objective function of Equation (7) is concave, and 2) otherwise, the objective function of Equation (7) is convex.* Detailed analysis can be found in Appendix Section A.

Based on the above conditions, we divide the possible situations into two subsets given the value of $k$ and provide closed-form solutions separately as follows. Its proof can be found in Appendix Section B.

*Theorem 1: The optimal data redemption amount for user i in Stage II is*

$$x_i^*(k) = \begin{cases} min\{max\{0, \dfrac{2d_i - kD}{2(1-k)}\}, d_i\}, & if\ 0 \le k < 1 \\ d_i, & otherwise. \end{cases} \tag{8}$$

## 3.2 Solutions of Stage I

We substitute the solution of $x_i$ in Theorem 1 into the objective function of Stage I (i.e., Equation (6)) and find the optimal $k^*$. The solution is not trivial because the objective function Equation (6) of Stage I after substituting $x_i$ is non-convex in $k$. However, we are able to provide a closed-form solution of the optimal unit price parameter by exploring two sub-problems. Specifically, we divide the problem according to different ranges of $k$ as

$$\max_{k \ge 0} \pi(k) = \max\{\max_{0 \le k < 1} \pi(k), \max_{k \ge 1} \pi(k)\}. \tag{9}$$

We find that we are either solving a concave problem (i.e., $\max_{0 \le k < 1} \pi(k)$) or a linear decreasing problem (i.e., $\max_{k \ge 1} \pi(k)$). We derive the following proposition and its proof can be found in Appendix Section C.

*Proposition 1:* $\max_{0 \le k < 1} \pi(k)$ *is a concave problem.*

The solution of $k$ is provided in the following Theorem 2. The proof can be found in Appendix Section C.

*Theorem 2: The optimal unit price parameter for the server is*

$$k^* = \begin{cases} max\{0, \dfrac{2p+q}{q}\}, & if\ \pi(max\{0, \dfrac{2p+q}{q}\}) > \pi(1) \\ 1, & otherwise, \end{cases} \tag{10}$$

*where*

$$p = (a\theta + t)(\sum_{i \in \mathcal{I}} d_i)^2 + \frac{1}{4}(\theta+1)I^2D^2 - \frac{1}{2}D\sum_{i \in \mathcal{I}} d_i + ID^2 + \sum_{i \in \mathcal{I}} d_i^2 \tag{11}$$

*and*

$$q = (-\theta I - 1)D\sum_{i \in \mathcal{I}} d_i - \frac{1}{2}(\theta+1)I^2D^2 - 2t(\sum_{i \in \mathcal{I}} d_i)^2 - 2ID^2 - \sum_{i \in \mathcal{I}} d_i^2. \tag{12}$$

# 4 PERFORMANCE EVALUATIONS

In the previous section, we derive the closed-form of the optimal unit price function setting for the server and the data redemption amount for each user. Given such optimal solutions, we are interested in some questions: *What types of situations are more beneficial for the server's profit? Does our mechanism result in a higher server's profit and users' payoff than other benchmark schemes? What is the major factor in our mechanism that caused the data removal cost?*

To answer the above questions, we conduct simulations to investigate the impact of parameter settings of the server's obtained data distribution in Section 4.1 and conduct experiments on two real datasets (i.e., MNIST [13] and Cifar-10 [20]) to prove the effectiveness of our solutions by comparing with three baselines in Section 4.2.

We obtain the following interesting insights:

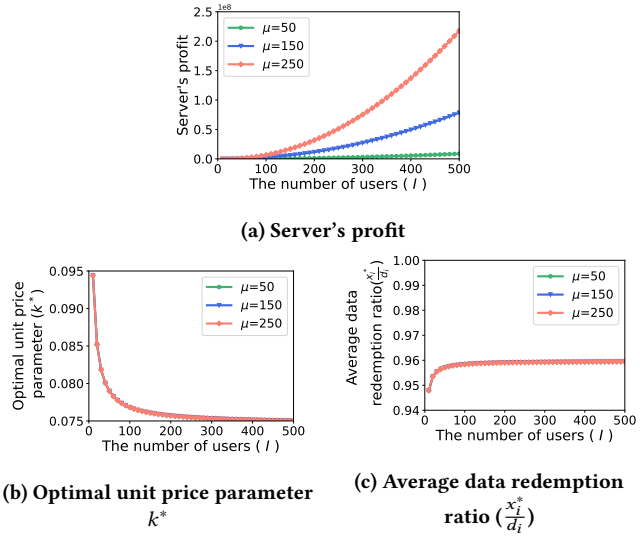
- Server can make more profit with a larger number of users and variation of the obtained data amount if it can optimize the unit price function accordingly.
- Server and users can obtain higher profit and payoff in our data redemption mechanism, which is a win-win for them.
- The data removal cost mostly happens in the accuracy cost if the server considers the . The server should decide whether to participate since the accuracy loss is significant.

## 4.1 Investigation of Parameters

We assume the amount of data $d_i$ obtained from user $i$ follows a Normal Distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu$ and $\sigma$ are the mean and the standard deviation of the amount of obtained data. Noted that the server has the full knowledge of the obtained data amount $\{d1, ..., d_I\}$ and therefore knows the mean $\mu$ and standard deviation $\sigma$ of the data. We adopt $D = 3 \cdot max\{d_1, ..., d_I\}$ and $\theta = 1$ in our simulations unless specified otherwise. For each set of parameter settings, we run the simulations 100 times with different random seeds and show the average value.

*4.1.1 The Impact of the Number of Users I.* We investigate the impact of the number of users $I$ on the server's profit under three settings of $\mu$. We set the $I$ in the range of 10 to 500 with a step size of 20. We set $\mu = 50, 100, 250$ to form three situations that contain various mean amounts of sold data. We set $\sigma = 0$ to eliminate the impact of data variation. The results are shown in Figure 4.

In Figure 4a, we plot the change of server's profit (i.e., Equation (6)) with respect to the increasing number of users $I$. We can

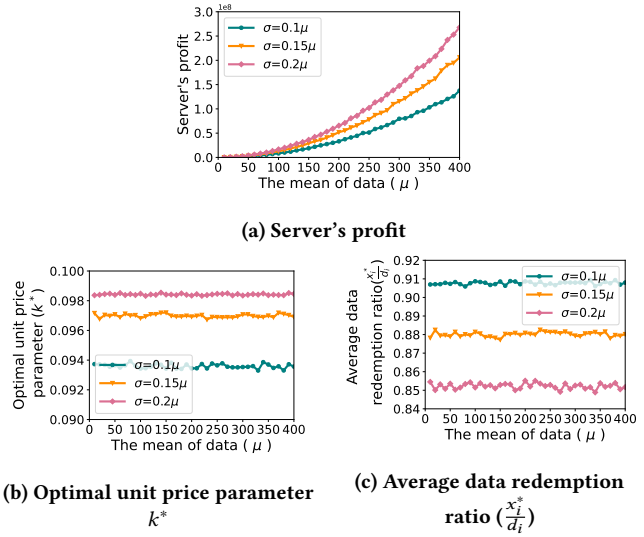



(a) Server's profit

(b) Optimal unit price parameter $k^*$

(c) Average data redemption ratio ($\frac{x_i^*}{d_i}$)

**Figure 4: Impact of the number of users ($I$).**




(a) Server's profit

(b) Optimal unit price parameter $k^*$

(c) Average data redemption ratio ($\frac{x_i^*}{d_i}$)

**Figure 5: Impact of the mean of data ($\mu$).**

observe that the profit of the server increases as the $I$ increases and the trend is strengthened with larger $\mu$. Specifically, the server that involves 500 users (i.e., $I = 500$) makes 32.86 times of profit that the server with only 100 users can make. Also, the server profit when $\mu = 250$ is almost 25 times larger than the server profit when $\mu = 50$ when both of them involve 500 users participating.

In Figure 4b, we plot the change of the optimal unit price function parameter (i.e., $k^*$) with respect to the increasing number of users $I$. We can observe that the server gradually decreases the value of $k^*$ as the number of users increases. The trend is aligned with our expectations because the compensation for unlearning can be allocated to more users and thus the price for each user can be less. We also find an interesting observation as below from the overlapped lines in Figure 4b. The proof of Observation 1 can be found in Appendix Section E.1.

*Observation 1: The mean of the obtained data $\mu$ has no impact on the optimal unit price function parameter $k^*$.*

In Figure 4c, we plot the change of the average data redemption ratio (i.e., $\frac{1}{I}\sum_{i\in\mathcal{I}}\frac{x_i^*}{d_i}$) with respect to the increasing number of users $I$. We can observe that the users increase their ratios of redemption data, which means the users are more willing to request more data back to gain a higher utility. Because the users observe a lower price (i.e., lower $k^*$) to redeem their data.

In summary, the server can decrease the unit price of compensation to make a larger profit.

*4.1.2 The Impact of the Mean Amount of Server's Obtained Data $\mu$.* We investigate the impact of the mean amount of obtained data $\mu$ on the server's profit under three settings of $\sigma$ in this section. We set $\mu$ in the range of 10 to 400 with a step size of 20. The standard deviation of the amount of obtained data is set to $\sigma = 0.1\mu, 0.15\mu, 0.2\mu$ to form three situations with various users' data deviations. The number of users is fixed to $I = 100$ to eliminate influence resulting from other factors. The results are shown in Figure 5.

In Figure 5a, we plot the change in the server's profit (i.e., $\pi(k)$ in Problem (6)) with respect to the increasing mean amount of obtained data $\mu$. We can observe that the server gains a higher level of profit when the mean amount of obtained data gets larger. The influence of $\mu$ is even more significant under a larger $\sigma$. In specific, in the situation $\sigma = 0.1\mu$, the server makes 3.97 times higher profit comparing two settings where $\mu = 200$ and $\mu = 400$, respectively. And we can see that as $\sigma$ increases from $0.1\mu$ to $0.2\mu$, the server's profit is dramatically increased 1.95 times when the mean amount of sold data $\mu$ is both equal to 300. We can summarize an interesting observation from the trend as follows. The proof of Observation 2 can be found in Appendix Section E.1.

*Observation 2: The server's profit increases with the mean amount of the server's obtained data $\mu$.*

In Figure 5b, we plot the change of the optimal unit price function parameter (i.e., $k^*$) with respect to the increasing mean amount of obtained data $\mu$. We can observe that the increment of $\mu$ has no impact on the value of $k^*$ in general, though a relatively large $\sigma$ results in a slight fluctuation of the value of $k^*$. This observation is in line with the above-mentioned Observation 1.

In Figure 5c, we plot the change of the average data redemption ratio (i.e., $\frac{1}{I}\sum_{i\in\mathcal{I}}\frac{x_i^*}{d_i}$) with respect to the increasing mean amount of obtained data $\mu$. We can observe that the average data redemption ratio keeps flat as the mean amount of obtained data increases, which means rational users choose to keep the same data redemption ratio. This is because the users receive a stable price from the server when $\mu$ is increasing.

*4.1.3 The Impact of the Standard Deviation of Server's Obtained Data Amount $\sigma$.* In this section, we investigate the impact of the standard variation $\sigma$ on the server's profit and users' payoff under three settings of $I$. The standard deviation of the obtained data amount is set to the range of 0 to 50 with a step size of 10. The number of users is set to $I = 10, 50, 100$ to form three situations. We set $\mu = 100$ to avoid influence from unrelated factors. The results are shown in Figure 6.

In Figure 6a, we plot the change of the optimal unit price function parameter (i.e., $k^*$) with respect to the increasing standard deviation of the obtained data amount $\sigma$. We can observe that $k^*$ increases and then decreases as $\sigma$ increases. This can be summarized as the following Observation 3 and its proof can be found in Appendix Section E.2.

*Observation 3: The increment of the standard deviation of the obtained data amount $\sigma$ results in an increasing and then decreasing trend on the optimal unit price function parameter $k^*$.*

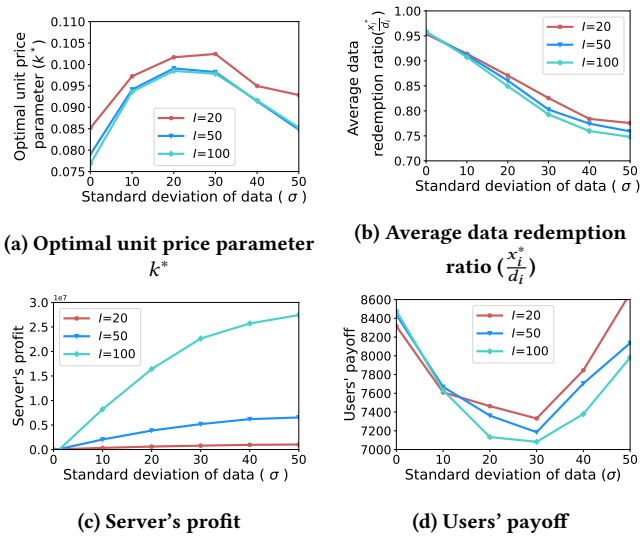
In Figure 6b, we plot the change of the users' average data redemption ratio (i.e., $\frac{1}{I}\sum_{i\in\mathcal{I}}\frac{x_i^*}{d_i}$) with respect to the increasing standard deviation of the obtained data amount $\sigma$. We can get an interesting observation as follows and it can be proved in Appendix Section E.2.

*Observation 4: The increment of the standard deviation of the obtained data amount $\sigma$ results in a decreasing trend on the users' average data redemption ratio.*

In Figure 6c, we plot the change in the server's profit (i.e., Equation (6)) with respect to the increasing standard deviation of the obtained data amount $\sigma$. We can observe that the server's profit is monotonically increasing with the increment of the standard deviation. And such a trend is significantly strengthened by a larger number of users $I$. Explicitly, with the same number of users (i.e., $I = 100$), the server gains 1.58 times of profit when the $\sigma$ is doubled from 20 to 40. Also, the profit made by the server with $I = 100$ is 4.21 times larger than the server with $I = 50$ under the same deviation (i.e., $\sigma = 50$). The reason why the increment of $\sigma$ can improve the server profit is the increment of $\sigma$ monotonously decreases the average data redemption ratio as mentioned in Observation 4. The fewer data users request back, the less unlearning cost the server needs to spend, which increases the profit significantly.

In Figure 6d, we plot the change of the users' average payoff (i.e., the average of Equation (7)) with respect to the increasing standard deviation of the obtained data amount $\sigma$. In general, the users' average payoff decreases at the beginning and then increases as the standard deviation becomes larger and larger. And the trend is more obvious with a larger number of users $I$. When the standard deviation is set to 10, the users' payoff is almost the same under three settings of $I$. When the standard deviation is raised to 50, the users' average payoff reaches a close level as the zero standard deviation situation. The explanation of such a trend can be found in the change of $k^*$. When $k^*$ increases at the beginning of the increment of $\sigma$, the payment charged from users are enlarged, which decreases the payoff of all users and discourages the users to request the data (results in a lower $\frac{x_i^*}{d_i}$). Later on, the value of $k^*$ drops, and the large $\sigma$ enlarges $D$ which can also degrade the payments. Therefore, the users' payoff raises again when $\sigma$ is relatively large.

From the above analysis, we can see that the optimal unit price function parameter $k^*$ decided by the server influences the decisions of users, and on the other side, the $x_i^*, i \in \mathcal{I}$ decided by users affects the server's profit in turn, which follows our two-stage formulation. What's more, we can obtain an insight that the server can make more profit with larger $\sigma$ if it can optimize the value of $k^*$ accordingly.




**(a) Optimal unit price parameter $k^*$**

**(b) Average data redemption ratio ($\frac{x_i^*}{d_i}$)**

**(c) Server's profit**

**(d) Users' payoff**

**Figure 6: Impact of the standard deviation of data ($\sigma$).**

## 4.2 Experiments on Real Datasets

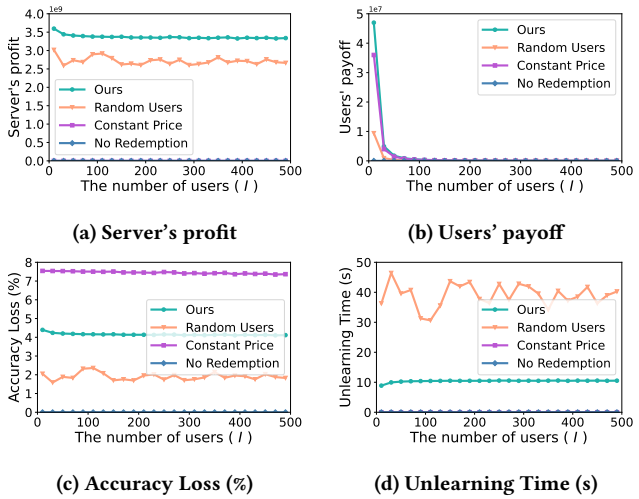
In this section, we conduct experiments on two popular real-world datasets (i.e., MNIST [13] and Cifar-10 [20]). We investigate the impact of the number of users $I$ on the server's profit, users' payoff, accuracy cost, and time cost in Section 4.2.2. We also compare them with three baselines to evaluate the effectiveness of our solutions.

*4.2.1 Experimental Settings.* For both of the datasets, we distribute the dataset to all users under an independent and identical distribution (IID). The MNIST dataset contains 60,000 images for training and Cifar-10 dataset contains 50,000 images for training. Specifically, we randomly split the whole dataset into many identical shards and distribute one shard to each user. Noted that the total number of data is the same no matter how many users participate in our setting. The parameter of the time and accuracy cost function $t$ and $a$ is obtained by the maximum likelihood estimation (MLE) of the results demonstrated in Figure 1 and Figure 2.

We set up the following three baselines to compare with.

- *Random Users*: In Stage II, the data redemption amount is set to $x_i = \lambda_i d_i$ where $\lambda_i, \forall i \in \mathcal{I}$ follow a normal distribution $\mathcal{N}(0,1)$. In Stage I, the server optimizes its unit price parameter $k$ by Theorem 2.
- *Constanct Price*: The unit price function is set to $P(x_i) \triangleq k, k \geq 0$. In Stage II, the data redemption amount is optimized to $x_i^* = max\{0, \frac{2d_i - k}{2}\}$; In Stage I, the unit price function parameter is optimized to $k^* = min\{\frac{[(a\theta+2t)I-1]\sum_{i\in\mathcal{I}} d_i}{\frac{1}{2}(a\theta+t)I^2+I}, 2d_i\}$ by plugging in the solution of Stage II. The proof of the solutions can be found in Appendix Section D.
- *No redemption*: In Stage II, the data redemption amount is set to $x_i = 0, \forall i \in \mathcal{I}$; In Stage I, the unit price setting follows Theorem 2.

We consider two costs as equally important (i.e., $\theta = 1$) and set the number of users $I$ in the range of 10 to 500 with a step size of 20. Other settings are the same as the settings in Section 4.1 unless mentioned.




(a) Server's profit (b) Users' payoff

(c) Accuracy Loss (%) (d) Unlearning Time (s)

**Figure 7: Impact of the number of users ($I$) on MNIST dataset.**

*4.2.2 The Impact of the Number of Users I.* The results on the MNIST dataset and Cifar-10 dataset are shown in Figure 7 and Appendix Figure 8, respectively. Since the results on the two datasets are similar, we discuss the results on the MNIST dataset in this section.

In Figure 7a, we plot the change of the server's profit (i.e., Equation (6)) with respect to the increasing number of users $I$. We can observe that the server's profit is generally stable in both our mechanism and other baselines when the number of users $I$ is increasing. It does not contradict our above-mentioned conclusions because the settings are different[4]. Besides, we can find that our mechanism performs better (gain the highest server's profit) than all baselines because Theorem 1 and 2 ensure the optimum of our solutions. Specifically, the server profit obtained by our mechanism is 1.23 times higher than the profit obtained by the *Random Users* baseline on average. The server can only gain very little profit under *Constant Price* baseline because the unit price $k^*$ is optimized to a very low value in this setting. Therefore, the users choose to redeem almost all of their data and the compensation is not sufficient for the server to gain profit. In the *No redemption* baseline, the server cannot gain profit because no compensation is received and no machine unlearning is performed.

In Figure 7b, we plot the change of the users' payoff (i.e., Equation (7)) with respect to the increasing number of users $I$. We can find that the users' payoff of our mechanism is higher than others, which proves that our mechanism can provide a win-win combination for both the server and users.

From these two figures, we can see that our mechanism can provide a win-win solution for both the server and users. We provide a higher profit for the server and a higher payoff for users compared with other baselines.

In Figure 7c, we plot the change of the accuracy loss in the percentage of all methods with respect to the increasing number of users $I$. We observe that our accuracy loss is only half of the loss of the *Constant Price* baseline on average. It is expected because a large amount of data is redeemed in *Constant Price* baseline based on the above analysis. On the contract, *Random Users* and *No Redemption* baselines suffer from much lower accuracy loss because there are only half of the data on average and none of the data is redeemed, respectively.

In Figure 7d, we plot the change of the unlearning time in the seconds of all methods with respect to the increasing number of users $I$. We can see that the time loss of *Constant Price* baseline is almost zero since the data redemption amount is large. Also, the time loss of *No Redemption* baseline is zero but the reason is different. It is because no machine unlearning is performed in *No Redemption* baseline.

From these two figures, we can notice that the accuracy loss is more significant than the unlearning time (e.g., $\sim$ 4.5% v.s. $\sim$ 10s) if the server considers the two costs equally important. Thus, we can summarize an interesting observation based on the results as follows.

*Observation 5: The data redemption cost of the server in our mechanism mostly happens in the accuracy loss if the server considers the two costs equally important.*

Therefore, the server should decide carefully whether it should participate in the data redemption mechanism. For example, the server should set a threshold of accuracy loss, if the threshold is exceeded, even though it can get a large compensation, the server should not participate.

## 5 CONCLUSION

In this paper, we studied the first incentive mechanism for data redemption on the basis of machine learning to the best of our knowledge. We provide sufficient compensation to the server and thus motivate it to perform machine unlearning effectively and honestly. We formulated a two-stage decision process for the server and users that maximize their profit independently. In our mechanism, the server decides its optimal unit price function parameter and the users decide the data redemption amount following the announced price. We derived the closed-form solutions for both sides by exploring the sub-problems even though the problems are non-convex. Experiment results on real-world datasets show that our mechanism can achieve a win-win combination for both the server and users compared with three baselines. We also found that the accuracy loss accounts for the majority of the machine unlearning cost.

**Online Resources**: Appendix Section D, E and F are available online at www. xxx.

## REFERENCES

[1] 2016. General Data Protection Regulation. https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng.
[2] 2018. California Consumer Privacy Act. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.
[3] 2018. PIPEDA privacy legislation. https://www.priv.gc.ca/en/opc-news/news-and-announcements/2018/an181010/.
[4] DP Bertsekas. 2003. Nonlinear Programming. 2nd edn. Athena Scientific, Belmont (1999) 8. Burer, S., Monteiro, RDC: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program* 95 (2003), 329357.
[5] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 141–159.

[4]The settings in Section 4.1 assume the same $\mu$ when $I$ increases, which means the total number of data is actually increasing. While in Section 4.2, the volume of data in the training set is fixed.

[6] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization.* Cambridge university press.
[7] Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*. IEEE, 463–480.
[8] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 896–911.
[9] Mia Xu Chen, Benjamin N Lee, Gagan Bansal, Yuan Cao, Shuyuan Zhang, Justin Lu, Jackie Tsay, Yinan Wang, Andrew M Dai, Zhifeng Chen, et al. 2019. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2287–2295.
[10] Valeriu Codreanu, Damian Podareanu, and Vikram Saletore. 2017. Scale out for large minibatch SGD: Residual network training on ImageNet-1K with improved accuracy and reduced time to train. *arXiv preprint arXiv:1711.04291* (2017).
[11] Maureen L Cropper, Leland B Deck, and Kenenth E McConnell. 1988. On the choice of funtional form for hedonic price functions. *The review of economics and statistics* (1988), 668–675.
[12] Yue Cui, Zhuohang Li, Luyang Liu, Jiaxin Zhang, and Jian Liu. 2022. Privacy-preserving Speech-based Depression Diagnosis via Federated Learning. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 1371–1374.
[13] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142.
[14] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*. PMLR, 2242–2251.
[15] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems* 32 (2019).
[16] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. 2021. Adaptive machine unlearning. *Advances in Neural Information Processing Systems* 34 (2021), 16319–16330.
[17] Ahsan Ilyas and Syed Shehryar Akbar. 2022. Machine Learning and Virtual Try on for Improving Sales and Purchase. *International Journal of Secure and Intelligent Computing (IJSIC)* 1, 1 (2022), 26–40.
[18] Indrawati Indrawati, Irmeilyana Irmeilyana, Fitri Maya Puspita, and Meiza Putri Lestari. 2014. Cobb-Douglass Utility Function in Optimizing the Internet Pricing Scheme Model. *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 12, 1 (2014), 227–240.
[19] Masayuki Karasuyama and Ichiro Takeuchi. 2009. Multiple incremental decremental learning of support vector machines. *Advances in neural information processing systems* 22 (2009).
[20] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[21] Hui Liu, Wenya Wang, and Haoliang Li. 2022. Towards Multi-Modal Sarcasm Detection via Hierarchical Congruity Modeling with Knowledge Enhancement. *arXiv preprint arXiv:2210.03501* (2022).
[22] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, et al. 2017. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442* (2017).
[23] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. 2022. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1749–1758.
[24] Badri Narayan Mohapatra and Prangya Prava Panda. 2019. Machine learning applications to smart city. *ACCENTS Transactions on Image Processing and Computer Vision* 5, 14 (2019), 1.
[25] Robinson Sitepu, Fitri Maya Puspita, Hadi Tanuji, and Icha Puspita Novyasti. 2016. Cobb-Douglas Utility Function Of Information Service Pricing Scheme Based On Monitoring And Marginal Costs. In *CONFERENCE PROCEEDING ICETS 2016*. 602.
[26] Cheng-Hao Tsai, Chieh-Yen Lin, and Chih-Jen Lin. 2014. Incremental and decremental training for linear classification. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 343–352.
[27] Yihao Xue, Chaoyue Niu, Zhenzhe Zheng, Shaojie Tang, Chengfei Lyu, Fan Wu, and Guihai Chen. 2021. Toward understanding the influence of individual clients in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 10560–10567.

# A THE CONVEXITY ANALYSIS OF THE OBJECTIVE FUNCTION OF EQUATION (7)

The objective function of Stage II in Equation (7) can be rewritten as

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad & f(x_i) = x_i P(x_i) - U_i(x_i) \\ & = (1-k)x_i^2 + kd_m x_i - 2d_i x_i, \\ \text{subject to} \quad & 0 \le x_i \le d_i. \end{aligned} \tag{13}$$

The second derivative of $f(x_i)$ is $f^{''}(x_i) = 2(1-k)$. Therefore, if $0 \le k < 1$, the $f(x_i)$ is a convex function because $f^{''}(x_i) \ge 0$, else $f(x_i)$ is a concave function. Since the objective function of Equation (7) is equal to $-f(x_i)$, we can get that when $0 \le k < 1$, the objective function of Equation (7) is concave, otherwise, the objective function of Equation (7) is convex.

# B PROOF OF THEOREM 1

We first provide the proof of Theorem 1 when the objective function of Equation (7) is concave in Appendix Section B.1 and then when it is convex in Appendix Section B.2.

## B.1 Concave Stage II

In our problem, the data redemption amount $x_i$ is constrained in $x_i \in [0, d_i]$, which means the variable that minimizes $f(x_i)$ may not occur in the domain of $x_i$ and makes the problem complex to solve. We can further divide the possible situations into three subsets according to the positional relationship between $\bar{x}$ that minimizes $f(x_i)$ and the boundary $[0, d_i]$ as follows. We define the conditions of each situation using the first derivative of the objective function of $f^{'}(x_i)$.

- *Situation A* ($0 < \bar{x} < d_i$): The conditions of this situation is $f^{'}(0) = kD - 2d_i < 0$ and $f^{'}(d_i) = kD - 2kd_i > 0$, which can be rewritten as $2kd_i < kD < 2d_i$.
- *Situation B* ($\bar{x} \le 0$): The conditions of this situation is $f^{'}(0) = kD - 2d_i \ge 0$, which can be rewritten as $kD \ge 2d_i$.
- *Situation C* ($\bar{x} \ge d_i$): The conditions of this situation is $f^{'}(d_i) = kD - 2kd_i \le 0$, which can be rewritten as $kD \le 2kd_i$.

Then we discuss the optimal data redemption amount in these three situations separately.

*B.1.1 $x_i$ in Situation A.* Let $\lambda_1, \lambda_2$ be the dual variables of the inequality constraints $x_i \ge 0$ and $x_i \le d_i$ in Equation (7). Since the objective function of Stage II is convex in Situation A, the Karush-Kuhn-Tucker (KKT) conditions [4] are both necessary and sufficient. We define the Lagrangian function $L(x_i, \lambda_1, \lambda_2)$ of Stage II as

$$L(x_i, \lambda_1, \lambda_2) = (1-k)x_i^2 + kd_m x_i - 2d_i x_i - \lambda_1 x_i + \lambda_2(x_i - d_i) \tag{14}$$

and its KKT conditions as

$$\begin{gathered} 0 \le x_i \le d_i, \\ \lambda_1, \lambda_2 \ge 0,\ -\lambda_1 x_i = 0,\ \lambda_2(x_i - d_i) = 0, \\ \nabla_{x_i} L = -2kx_i + kD + 2x_i - 2d_i - \lambda_1 + \lambda_2 = 0. \end{gathered} \tag{15}$$

Recall that Situation A happens when $0 \le \bar{x} \le d_i$, we can let $\lambda_1^*, \lambda_2^* = 0$ and derive the solution of $x_i$ given $k$ as

$$x_i^* = \frac{2d_i - kD}{2(1-k)}. \tag{16}$$

We can prove the validity of $x_i^*$ by the following two perspectives.

- *Proof of* $x_i^* > 0$. In Situation I-A, we have a condition $kD - 2d_i < 0$, which is equivalent to $2d_i - kD > 0$. Given $0 \le k < 1$, we can get $x_i^* = \frac{2d_i - kD}{2(1-k)} > 0$.
- *Proof of* $x_i^* < d_i$. Given $d_i < D$ and $0 \le k < 1$, we have $k(d_i - D) < 0$. We can derive $2d_i - kD - 2d_i(1-k) < 0$ and then $x_i^* = \frac{2d_i - kD}{2(1-k)} < d_i$ is proved.

*B.1.2 $x_i$ in Situation B.* Situation B happens when the minimal point $\bar{x} \le 0$, then the optimal $x_i$ is its minimum value that $x_i^* = 0$.

*B.1.3 $x_i$ in Situation C.* Situation I-C happens when the minimal point $\bar{x} \ge d_i$, then the optimal $x_i$ is its maximum value that $x_i^* = d_i$.

From the above Appendix Section B.1.1, Section B.1.2 and Section B.1.3, we can combine the solutions into a united form that

$$x_i^* = min\{max\{0, \frac{2d_i - kD}{2(1-k)}\}, d_i\}, \ if\ 0 \le k < 1. \tag{17}$$

## B.2 Convex Stage II

When the objective function of Stage II is convex, the optimal $x_i^*$ is one of the boundaries which makes the objective function larger. We can compute the results for both boundaries of its opposite function $f(x_i)$ by

$$f(0) = 0, \ f(d_i) = (-1-k)d_i^2 + kd_iD. \tag{18}$$

Since $D > d_i$, we can further enlarge the $f(d_i)$ by

$$f(d_i) = (-1-k)d_i^2 + kd_iD < (-1-k)d_iD + kd_iD = -d_iD < 0. \tag{19}$$

Then we can see $f(d_i) < f(0)$. We can derive that the optimal solution is

$$x_i^* = d_i, \ if\ k > 1. \tag{20}$$

# C PROOF OF THEOREM 2

The objective function of Stage I in Equation (6) can be rewritten as follows.

$$\begin{aligned} \underset{k}{\text{minimize}} \quad g(k) \quad &= C(\textstyle\sum_{i\in\mathcal{I}} x_i) - \sum_{i\in\mathcal{I}} x_i P(x_i) \\ &= (a\theta + t)(\textstyle\sum_{i\in\mathcal{I}} x_i)^2 - 2t\sum_{i\in\mathcal{I}} d_i \sum_{i\in\mathcal{I}} x_i \\ &\quad + t(\textstyle\sum_{i\in\mathcal{I}} d_i)^2 + k(\sum_{i\in\mathcal{I}} x_i^2 - d_m \sum_{i\in\mathcal{I}} x_i), \\ \text{subject to} \quad &\quad k > 0. \end{aligned} \tag{21}$$

Given the non-convexity of Equation (21), we explore the convexity of $g(k)$ in different ranges of $k$ and find that we are either solving a convex or a linear increasing problem. We derive their optimal solutions separately in Appendix Section C.1 and Appendix Section C.2.

## C.1 Situation i: $0 \le k < 1$

By replacing the $x_i$ in $g(k)$ according to Theorem 1, we can rewrite Equation (21) as

$$\begin{aligned} \underset{k}{\text{minimize}} \quad g(k) = \quad &[(a\theta + t)(\textstyle\sum_{i\in\mathcal{I}} d_i)]^2 \frac{1}{(1-k)^2} \\ &+ [(-\theta I - 2)D\textstyle\sum_{i\in\mathcal{I}} d_i] \frac{k}{(1-k)} \\ &+ [\frac{(\theta+1)I^2D^2 - 2D\sum_{i\in\mathcal{I}} d_i}{4}] \frac{k^2}{(1-k)^2} \\ &+ [-2t(\textstyle\sum_{i\in\mathcal{I}} d_i)^2] \frac{1}{(1-k)} \\ &+ [ID^2] \frac{k^2}{(1-k)} + [\textstyle\sum_{i\in\mathcal{I}} d_i^2] \frac{k}{(1-k)^2} \\ &+ [ID^2] \frac{k^3}{(1-k)^2} + t(\textstyle\sum_{i\in\mathcal{I}} d_i)^2, \\ \text{subject to} \quad &0 \le k < 1. \end{aligned} \tag{22}$$

Let $\gamma(k) = \frac{1}{1-k}, \gamma \ge 1$. We can find that $\gamma$ is convex in $k$ by its secondary condition where $\gamma'' = \frac{2}{(1-k)^3} > 0$. Then, we can rewrite Equation (22) as

$$\begin{aligned} \underset{\gamma}{\text{minimize}} \quad &\beta(\gamma) = p\gamma^2 + q\gamma + r, \\ \text{subject to} \quad &\gamma \ge 1, \end{aligned} \tag{23}$$

where $p, q$ are defined in Equation (11), Equation (12) and

$$r = (\theta I + \frac{3}{2})D\sum_{i\in\mathcal{I}} d_i + \frac{1}{4}(\theta+1)I^2D^2 + ID^2 + t(\sum_{i\in\mathcal{I}} d_i)^2. \tag{24}$$

We can observe that $q < 0$ from its equation because all items are negative. Similarly, all items in $r$ are positive, therefore $r > 0$. Because $\sum_{i\in\mathcal{I}} d_i < ID$, we can prove $p > 0$ by replacing the only negative item in $p$ as

$$\begin{aligned} p > \quad &(a\theta + t)(\textstyle\sum_{i\in\mathcal{I}} d_i)^2 + \frac{1}{4}(\theta+1)I^2D^2 - \frac{1}{2}D \cdot ID + ID^2 + \sum_{i\in\mathcal{I}} d_i^2 \\ = \quad &(a\theta + t)(\textstyle\sum_{i\in\mathcal{I}} d_i)^2 + \frac{1}{4}(\theta+1)I^2D^2 + \frac{1}{2}ID^2 + \sum_{i\in\mathcal{I}} d_i^2 > 0. \end{aligned} \tag{25}$$

Based on the above analysis, $\beta(\gamma)$ is a standard quadratic function, where $p > 0, q < 0$, and $r > 0$.

Thus, we can find $g(\gamma(k))$ is a convex problem [6] in $k$ because $\beta(\gamma)$ is nondecreasing convex and $\gamma(k)$ is convex. We can prove Proposition 1 that $\pi(k)$ is concave since $\pi(k) = -g(k)$.

Let $\bar{\gamma} = argmin\ \beta(\gamma)$, we can compute $\bar{\gamma} = \frac{q}{-2p}$ by the symmetry of $\beta(\gamma)$. According to the positional relationship between $\bar{\gamma}$ and the boundary of $\gamma$ (i.e., $\gamma \ge 1$), Situation i can also be divided into two subsets: 1) $\bar{\gamma} > 1$ and 2) $\bar{\gamma} \le 1$.

In the former subset, the $\bar{\gamma}$ that minimizes the $\beta(\gamma)$ exists within the boundary of $\gamma$, therefore, $\gamma^* = \bar{\gamma} = \frac{q}{-2p}$. By the relationship between $k$ and $\gamma$, we can get $k^* = \frac{2p+q}{q}$. The validity of $k^*$ can be proved as follows.

- *Proof of* $k^* > 0$. In this subset of Situation I-A, we have $\frac{q}{-2p} > 1$, which can derive $2p + q < 0$. Given $q < 0$, we have $k^* = \frac{2p+q}{q} < 0$.
- *Proof of* $k^* < 1$. $k^* = \frac{2p+q}{q} = \frac{2p}{q} + 1 < 1$ since $p > 0$ and $q < 0$.

In the latter subset where $\bar{\gamma}$ is on the left side of the boundary of $\gamma$, we can directly derive that the optimal $\gamma$ is its minimum value that $\gamma^* = 1$. Thus, we can get $k^* = 0$ in this subset.

We also notice that in the latter subset where $\bar{\gamma} \le 1$, we can compute that $\frac{2p+q}{q} < 0$, thus we can summarize the solution of the optimal unit price parameter in a united form by

$$k^* = max\{0, \frac{2p+q}{q}\}, if\ 0 \le k < 1. \tag{26}$$

## C.2 Situation ii: $k \ge 1$

When $k \ge 1$, we can use the solution provided by Theorem 1 (i.e., $x_i^* = d_i$) to replace $x_i$ in $g(k)$ and rewrite Equation (21) as

$$\underset{k}{\text{minimize}} \quad g(k) = \quad (D\textstyle\sum_{i\in\mathcal{I}} d_i - \sum_{i\in\mathcal{I}} d_i^2)k + (\theta - 1)(\sum_{i\in\mathcal{I}} d_i)^2, \tag{27}$$

Because $D > d_i$, we can derive that $D\sum_{i\in\mathcal{I}} d_i > \sum_{i\in\mathcal{I}} d_i^2$. Obviously, the above function is a linear increasing function of $k$. Then the optimal $k$ is the minimum value which is $k^* = 1$.

# D SOLUTION OF CONSTANT PRICE BASELINE

In the *Constant Price* baseline, let $P(x_i) \triangleq k, k \geq 0$ be the unit price function. Similarly, we also solve the profit maximization problem of the server and the payoff maximization problem of users individually.

## D.1 Solution of Stage II in Constant Price

We can write the objective function of Stage II in the Constant Price baseline as

$$\begin{aligned} \underset{x_i}{\text{minimize}} \quad h(x_i) \quad &= x_i P(x_i) - U_i(x_i) \\ &= x_i^2 + (k - 2d_i)x_i, \\ \text{subject to} \quad &0 \leq x_i \leq d_i. \end{aligned} \tag{28}$$

We can find that $h(x_i)$ is a convex problem by computing its secondary derivate $h^{''}(x_i) = 2 > 0$. Also, we can find that $h(x_i)$ is a quadratic function, thus we can derive the solution of the optimal data redemption amount as

$$x_i^* = min\{max\{0, \frac{2d_i - k}{2}\}, d_i\}. \tag{29}$$

Since the constraint of the unit price function is $k \geq 0$, we can derive that $\frac{2d_i - k}{2} \leq d_i$. Then we can rewrite the solution of Stage II in a simpler way that

$$x_i^* = max\{0, \frac{2d_i - k}{2}\}. \tag{30}$$

## D.2 Solution of Stage I in Constant Price

The objective function of Stage I can be rewritten as

$$\begin{aligned} \underset{k}{\text{minimize}} \quad z(k) \quad &= C(\textstyle\sum_{i\in\mathcal{I}} x_i) - \sum_{i\in\mathcal{I}} x_i P(x_i) \\ &= (a\theta + t)(\textstyle\sum_{i\in\mathcal{I}} x_i)^2 - 2t \sum_{i\in\mathcal{I}} d_i \sum_{i\in\mathcal{I}} x_i \\ &\quad + t(\textstyle\sum_{i\in\mathcal{I}} d_i)^2 - k(\sum_{i\in\mathcal{I}} x_i^2 \\ \text{subject to} \quad &k \geq 0. \end{aligned} \tag{31}$$

We replace the $x_i$ in Equation (31) by Equation (30), and similarly, we also explore two sub-problem by different ranges of $k$.

When $k \geq 2d_i$, the optimal solution of Stage II is $x_i^* = 0$. Then the objective function of Stage I is a constant where

$$z(k) = t(\sum_{i\in\mathcal{I}} d_i)^2, if\ k \geq 2d_i. \tag{32}$$

In such case, the value of $k$ is meaningless because no data will be redeemed.

When $0 \leq k < 2d_i$, the optimal solution of Stage II is $x_i^* = \frac{2d_i - k}{2}$. Then the objective function of Stage I can be rewritten as

$$\begin{aligned} z(k) = \quad &(\tfrac{1}{4}(a\theta + t)I^2 + \tfrac{1}{2}I)k^2 - (((a\theta + 2t)I - 1)\textstyle\sum_{i\in\mathcal{I}} d_i)k \\ &+ a\theta(\textstyle\sum_{i\in\mathcal{I}} d_i)^2,\ if\ 0 \leq k < 2d_i. \end{aligned} \tag{33}$$

The Equation (33) can be considered as a quadratic function

$$z(k) = uk^2 + vk + w, \tag{34}$$

where $u > 0, v < 0, w > 0$. Given the symmetry of Equation (33), let $\bar{k} = \frac{v}{-2u} > 0$ be the symmetry axis that minimizes $z(k)$. Considering the positional relationship between $\bar{k}$ and its boundary $2d_i$ in this situation, we can derive the optimal constant price parameter as

$$k = min\{\frac{v}{-2u}, 2d_i\}. \tag{35}$$

Combing the above two situations, the optimal constant price parameter is

$$k = min\{\frac{[(a\theta + 2t)I - 1]\sum_{i\in\mathcal{I}} d_i}{\frac{1}{2}(a\theta + t)I^2 + I}, 2d_i\}. \tag{36}$$

# E PROOF OF OBSERVATIONS

## E.1 Proof of Observation 1&2

In this section, we provide a case study to prove two observations: 1) *Observations 1 ($\mu$ and the value of $k$)*: the changes in the mean amount of sold data $\mu$ do not affect the value of $k$; and 2) *Observations 2 ($\mu$ and the server profit)*: the mean amount of sold data $\mu$ has positive influences on the server's profit ($-g(k)$).

Consider a case where $\mu = d$. Following the setting in Section 4.1.3, we have $\sigma = 0$, $D = 3d$ and $\theta = 1$. Thus, we can derive that $\sum_{i\in\mathcal{I}} d_i = Id$, $\sum_{i\in\mathcal{I}} d_i^2 = Id^2$ and $D^2 = 9d^2$.

Using the solution of $k$ in Theorem 2, we can derive that

$$\begin{aligned} k^* = \quad &\frac{2[(a\theta+t)(\sum_{i\in\mathcal{I}} d_i)^2 + \frac{1}{4}(\theta+1)I^2D^2 - \frac{1}{2}D\sum_{i\in\mathcal{I}} d_i + ID^2 + \sum_{i\in\mathcal{I}} d_i^2]}{(-\theta I-1)D\sum_{i\in\mathcal{I}} d_i - \frac{1}{2}(\theta+1)I^2D^2 - 2t(\sum_{i\in\mathcal{I}} d_i)^2 - 2ID^2 - \sum_{i\in\mathcal{I}} d_i^2} + 1 \\ = \quad &\frac{2[2I^2d^2 + 13Id^2]}{-14I^2d^2 - 22Id^2} = \frac{-10I+4}{-14I-22}, \end{aligned} \tag{37}$$

which demonstrates that the value of $k^*$ is not affected by $\mu$.

Since the average requested data ration $\frac{x_i}{d_i}$ is stable with $\mu$, we can simply set $x_i = \gamma d_i$ and therefore, $\sum_{i\in\mathcal{I}} x_i = \gamma \sum_{i\in\mathcal{I}} d_i$. Using Equation (21), we can derive that the server profit is

$$-g(k) = 2\gamma^2 I^2 d^2 - 2\gamma I^2 d^2 + k(3\gamma Id^2 + \gamma^2 Id^2), \tag{38}$$

which is a monotonically increasing function of $d$ (i.e., $\mu$) when the value of $k$ is stable.

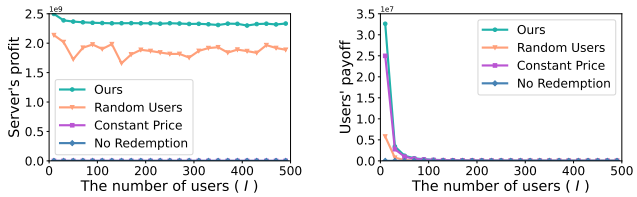
## E.2 Proof of Observation 3&4

In this section, we provide a case study to prove two observations: 1) *Observations 3 ($\sigma$ and the value of $k$)*: the increment of the standard deviation of the sold data $\sigma$ results in an increasing and then decreasing trend on the value of $k$; and 2) *Observations 4 ($\sigma$ and the requested ratio $\frac{x_i}{d_i}$)*: the increment of the standard deviation of the sold data $\sigma$ monotonically decreases the requested data ratio.

Similar to Appendix Section E.1, consider a case where $\mu = d$. Following the setting in Section 4.1.3, we have $\sigma = 0.1d$, $a = 1$, and $t = 1$. Thus, we can derive that $\sum_{i\in\mathcal{I}} d_i = Id$ and $\sum_{i\in\mathcal{I}} d_i^2 = Id^2$.

Using the solution of $k$ in Theorem 2, we can derive that

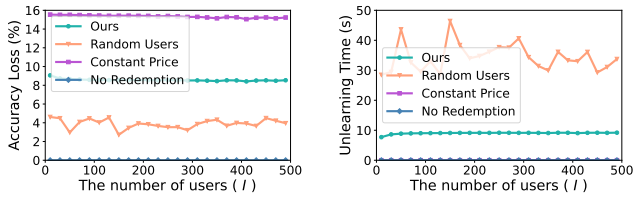
$$\begin{aligned} k = \quad &\frac{2[(a\theta+t)(\sum_{i\in\mathcal{I}} d_i)^2 + \frac{1}{4}(\theta+1)I^2D^2 - \frac{1}{2}D\sum_{i\in\mathcal{I}} d_i + ID^2 + \sum_{i\in\mathcal{I}} d_i^2]}{(-\theta I-1)D\sum_{i\in\mathcal{I}} d_i - \frac{1}{2}(\theta+1)I^2D^2 - 2t(\sum_{i\in\mathcal{I}} d_i)^2 - 2ID^2 - \sum_{i\in\mathcal{I}} d_i^2} + 1 \\ = \quad &\frac{D(3D-d) + 4Id^2 + 2d^2}{-D(Id+d+3ID) - 2Id^2 - d^2}. \end{aligned} \tag{39}$$




(a) Server's profit




(b) Users' payoff




(c) Accuracy Loss (%)




(d) Unlearning Time (s)

**Figure 8: Impact of the number of users ($I$) on Cifar-10 dataset.**

Let $A = Id + d > 0$, $B = 4Id^2 + 2d^2 > 0$, and $C = -2Id^2 - d^2 < 0$, which are all constants. We can rewrite the value of k by

$$k = \frac{D(3D - d) + B}{-D(3ID + A) + C}, \tag{40}$$

which is related to $D$. And the increment of $\sigma$ results in a monotonical increment of $D$. Therefore, the relationship between $\sigma$ and $k$ is equalized to the relationship between $D$ and $k$.

To further analyze $k$, we compute its first-order derivative

$$k' = \frac{(-3Id - 3A)D^2 + (Ad + 6C + 6IB)D - Cd + AB - Ad}{(-D(3ID + A) + C)^2}, \tag{41}$$

where $-3Id - 3A < 0$, $Ad + 6C + 6IB > 0$, and $-Cd + AB - Ad > 0$. Therefore, we can find that $k'$ is positive and then becomes negative according to the properties of quadratic functions. Then we can derive that the value of $k$ is first increasing and then decreasing as the increment of $\sigma$.

Then, the analysis of $x_i$ can be divided into two situations: 1) $k$ increases, $\sigma$ increases and 2) $k$ decreases, $\sigma$ increases.

In the first situation, using Equation (16), we can derive

$$x_i = \frac{2d_i - kD}{2(1-k)} = \frac{d_i}{1-k} - \frac{D}{2(\frac{1}{k} - 1)}, \tag{42}$$

which is simply decreasing in this situation.

While in the second situation, the increment of $D$ is much more significant than the decrement of $k$, because any slight increment of $\sigma$ results in a dramatical increment of $max\{d_1, ..., d_I\}$. Thus, $kD$ is still increasing when $\sigma$ increases and $x_i = \frac{2d_i - kD}{2(1-k)}$ is also decreasing.

## F RESULTS ON CIFAR-10 DATASET

We also investigate the impact of the number of users $I$ on the Cifar-10 dataset. The results are shown in Figure 8. We can observe that the results are similar to the MNIST dataset, thus related analysis can be found in Section 4.2.