# Differential Privacy in Machine Learning - A Comparative Study

Inderjot Kaur Ratol
McGill ID: 260661928
inderjot.ratol@mail.mcgill.ca

Nirmal
McGill ID: 260716737
nirmal.kanagasabai@mail.mcgill.ca

Taanvi Chhetri
McGill ID: 260567764
taanvi.chhetri@mail.mcgill.ca

***Abstract*—Privacy-preservation in machine learning models is one of the most active research areas in the field of machine learning. In the recent years, many researchers have started designing new techniques to preserve the privacy of machine learning models. Differential privacy is the most widely used privacy preservation technique which provides strong privacy guarantees and is applicable to wide range of datasets. In this paper, we study two different techniques of applying differential privacy on the machine learning models- *Differential Privacy with Stochastic Gradient Descent in Deep Learning* and *Differential Privacy with Multiple Teachers and Students*. Then, we conduct a comparative study of these two techniques with a common dataset, i.e. MNIST, mentioned in the respective papers. Furthermore, we extend the *Multiple Teachers* technique with CIFAR-10 dataset and compare the performance with *Stochastic Gradient Descent* technique. Finally, we show the results of comparison between these two techniques followed by the discussion about which approach is better at preserving privacy.**

## I. Introduction

With the advent of technology, the amount of digital data generated every day has increased exponentially. The abundance of data has lead to the rise in the curiosity of analyzing this data and capturing the useful information that can be utilized in numerous ways in order to take the existing technology to an unprecedented level. Machine Learning is widely used to analyze this data. In the past few years, average people have accepted the idea of sending a lot of personal information to the various services they use. Surveys tell us that people are starting to feel uncomfortable about sending their personal information to these services [4]. Other than gaining access to the personal information through the services used by public, many companies have started collecting data by alternative methods like crowd-sourcing, i.e. gathering the data by involving public to take the questionnaire and collect the data. And, the data collected by the companies is used to market their products. The purpose of data collection is definitely to improve the services provided on phones, laptops etc. Many times, the data collected contains sensitive information about the user and the data collected is further used to train the machine learning models which are then in turn used to improve the services provided. In this paper, we describe how the machine learning models pose a threat of invading private information used to train these models.

This paper is divided into the following sections - Section II describes how the privacy preservation is related to machine learning models and how the techniques, *Differential Privacy with Stochastic Gradient Descent in Deep Learning* [8] and *Differential Privacy with Multiple Teachers and Students* [9], attempt to solve the problem. Section III discusses the meaning and the general model of *differential privacy*. Section IV-B and IV-C describe in the detail the two different techniques of differential privacy studied as part of this project.Section VI provides an extensive study of the various hyper parameter and the optimal values found for each parameter and provides a comparative view of these two techniques along with the results obtained by extending these techniques with new datasets. Section VII focuses on reviewing the existing literature related to privacy preservation and different techniques used to achieve the preservation.

## II. Problem Statement

Previously, when machine learning techniques were in their development phase, privacy preservation of the data used to train the models was not a concern. But, with the increasing threats of privacy breach in these models, for instance *Facebook's* face recognition algorithm, have called for the mandatory steps to be taken for preservation of the sensitive information [6]. Although, machine learning models are trained not to overfit the data but these models tend to memorize the features they are trained on and therefore leave an open window for the adversaries to attack and obtain the sensitive information. Fredrikson et al. demonstrated a model-inversion attack that recovers images from a facial recognition system [7]. Model-inversion attacks only require "black-box" techniques to attack a trained model in order to obtain the learned parameters which subsequently provides access to sensitive information.

To tackle this problem, many different techniques have been developed to preserve the private information in machine learning. These techniques include manipulation of the raw data by using numerous techniques like randomization of the data, generalizing the data etc. [1] Generalizing the data means losing some part of the original information. However, it helps the machine learning models to preserve privacy. The data manipulation methods are applicable to the datasets with text, numerical data but cannot be directly applied to datasets containing images. The need of privacy preservation in images

[1] http://www.cs.mcgill.ca/~jpineau/comp551/Lectures/BenjaminFung-presentation.pdf

lead to the development of new and reliable method of privacy preservation. This method is called *Differential Privacy* [5].

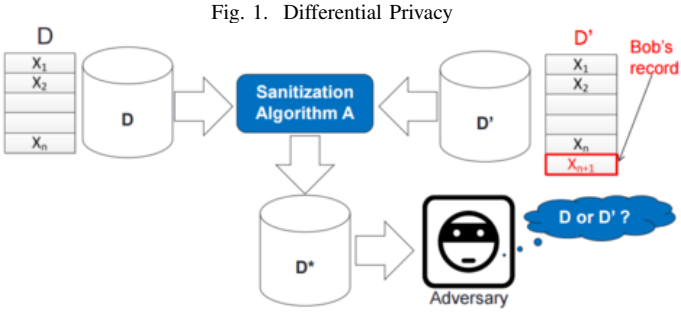
## III. Differential Privacy

Before detailing the techniques studied and compared in this paper, it is pertinent to discuss the meaning of differential privacy.

**Definition**- Consider two otherwise identical databases (D and D'), one with your information in it, and one without it. Differential Privacy ensures that the probability of a statistical query will produce a result is (nearly) the same whether its conducted on the first (D) or second database (D*). For instance, consider the training dataset is a set of image-label pairs. The two datasets are adjacent if they differ in a single entry, that is, if one image-label pair is present in one set and absent in the other.

Figure 1 shows the original database (D), i.e. the database with sensitive information. D' is the database which does not contain all the records present in database (D). Conceptually, a new database (D*) is created by merging the D and D' and machine learning model is trained on the inputs taken from this database, i.e. D*. Having the database tempered with additional information, the adversary querying database (D*) cannot figure out the source of data. This helps the machine learning models to preserve privacy by preventing the adversary from obtaining any sensitive information. In mathematical terms, it can be defined as- A randomized mechanism $M : D \rightarrow R$ with domain $D$ and range $R$ satisfies $(\epsilon, \delta)$-differential privacy if for any two adjacent inputs $d, d' \in D$ and for any subset of outputs $S \subseteq R$ it holds that

Fig. 1. Differential Privacy




$$Pr[M(d) \in S] \leq e^{\epsilon} Pr[M(d') \in S] + \delta$$

Dwork et al. [19] introduced $\delta$ which allows for the possibility that plain $\epsilon$ -differential privacy is broken with probability $\delta$. A common paradigm for approximating a deterministic real-valued function$f : D \rightarrow R$ with a differentially private mechanism is via additive noise calibrated to f's sensitivity $S_f$, which is defined as the maximum of the absolute distance $|f(d) - f(d')|$ where d and d' are adjacent inputs [8].

*Differential Privacy* is widely used technique to preserve privacy in the field of machine learning. It constitutes a standard for privacy preservation and provides strong privacy guarantees on application-specific databases.

## IV. Methodology

### A. Dataset and Feature Selection

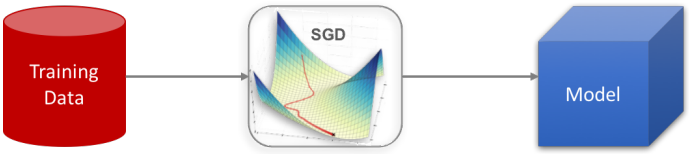
The datasets used to test the approaches are MNIST and CIFAR-10. MNIST is the standard dataset for handwritten digits recognition, where each image in the dataset is a 28*28 size gray-level image. The dataset is divided into two parts- 60000 images are used for training the algorithm while 10000 images are used in the testing phase [14]. CIFAR-10 dataset consists of colored images classified into 10 classes partitioned into 50000 for training and 10000 for testing of the algorithm [1]. Each image is a 32*32 image with 3 RGB channels. We have not applied any other preprocessing techniques in order to keep the implementation and testing consistent with the original papers.

### B. Approach I: Differential Privacy with SGD

*1) Implementation:* In the recent years, Neural networks have become one of the most powerful classifiers and perform remarkably well on complex objectives. They are used for many different types of machine learning problems like image classification, text representation and classification etc. *Differential Privacy with Stochastic Gradient Descent* (henceforth DP-SGD) makes use of the state-of-art techniques, in this case Neural Networks, combined with advanced privacy preservation mechanisms to protect the private data. DP-SGD follows the baseline model of Feed Forward Neural Networks (FFNN) and modifies the process of back-propagation (explained under the subsection IV-B2), i.e. the part using SGD, to apply privacy preservation techniques. Figure 2 shows the basic machine learning model training with Stochastic Gradient Descent. Here the training data is used to feed the neural network that uses stochastic gradient descent to back propagate the weights in order to get the optimal values of weights which further reduce the loss of classifier. At the end of this process, we are left with a trained machine learning model, in this case neural networks, which tends to make minimalistic mistakes while classifying or predicting the desired output. For MNIST dataset, a simple feed forward neural network with *ReLU* units and *softmax* of 10 classes (corresponding to the 10 digits) is used with cross-entropy loss and an optional PCA input layer. For CIFAR-10 dataset, Google's Tensorflow based implementation of Deep CNN is used for classification purposes [2].The network architecture consists of two convolutional layers followed by two fully connected layers. The convolutional layers use 5*5 convolutions with stride 1, followed by a ReLU and 2*2 max pools, with 64 channels each. Thus, the first convolution outputs a 12*12*64 tensor for each image, and the second outputs a 6*6*64 tensor [8]. Other than introducing two different way of implementing differential privacy with SGD , which are discussed later in section IV-B2, one of the main contributions is the development of a tool called *Moments Accountant*.

*Moments Accountant:* For DP-SGD, an important task is to compute the privacy loss incurred during the training phase which is done with the help of *Privacy accountant* [17]. For a single training instance, the privacy cost is defined by $|f(d) - f(d')|$ where d and d' are adjacent inputs, described in section III. Abadi et. al introduced a new tool for accounting privacy cost, called *Moments accountant*, which is built upon the already existing *Privacy accountant* and provides much tighter bounds on the on the composite privacy loss. The authors of DP-SGD claim that moments accountant is better than strong composition theorem which is known to provide best overall bound [8], [16].

Fig. 2. Baseline SGD model

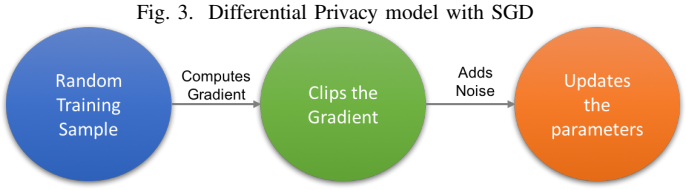



*2) Training Procedure:* There are three two components introduced by the authors of DP-SGD in the training of models used which are discussed below:

*a) Adding Noise to the gradient:* During the training phase, i.e. stochastic gradient descent is applied, DP-SGD clips the gradient per batch and adds noise to every training instance in that batch, where the noise clipped is equal to the $l_2$ norm vector of gradient per batch. It uses *lots* instead of batches, where a lot consists of one or more batches. The gradient is replaced by $g = max(1, \frac{||g||^2}{C})$, where C is the clipping threshold. Figure 3 shows a high-level view of how differential privacy, i.e. by adding noise, is combined with the standard SGD.

Adding noise to each training sample is claimed to be sophisticated way of introducing differential privacy because it controls the influence of training data during the training process.

Fig. 3. Differential Privacy model with SGD




*b) Noisy PCA:* Abadi et. al incorporated PCA in the neural network architecture used in the model. They claimed that applying PCA resulted in reducing the training time and improved accuracy by 2%. In order to make PCA differential private, they introduced noise into the PCA projections [15]. The authors of the DP-SGD followed the same approach used by Dwork et. al., i.e. by taking a random sample of training examples, normalizing each vector to $l_2$ norm and adding Gaussian noise to the co-variance matrix. They applied PCA projection over each input before they fed the inputs to neural networks. It ensures that the differential privacy is maintained with/without the use of PCA for neural networks.

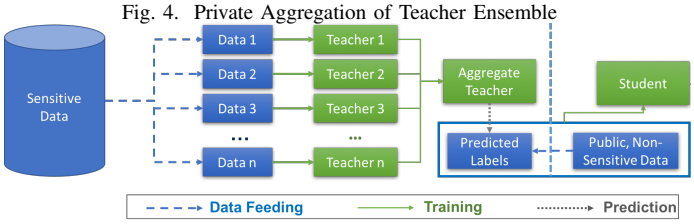
### C. Approach II : Multiple Teachers and Students

*1) Implementation:* Machine Learning algorithms, however robust they are, can be reverse-engineered and made to expose private and sensitive user data. In order to learn more about a particular model, repetitive queries are made on the opaque model by the adversary until he retrieves the information required. Also, the privacy guarantees do not hold when the adversary has access to the internal model parameters.

To address this issue, two different models say, 'teachers' and 'students' are used. The proposed approach, Private Aggregation of Teacher Ensembles (PATE), operates in a 'black-box' manner where multiple teacher models are trained on disjoint datasets and are combined to perform the predictions. It offers strong privacy guarantees for the training data by not publishing the teachers and using them with the non-private data to train the student model. The student predicts an output that is chosen by noisy voting among all of the teachers. Also, the student cannot access an individual teacher or the underlying data or its parameters. Doing this makes the student immune not only to queries made by an adversary but also to the inspection being made on its internal workings.

Figure 4 shows the PATE approach which begins by splitting the sensitive user data into disjoint subsets. An ensemble of teacher models [18] is used to train on the chunks of data. The aggregated output of the ensemble is collected and is coupled with auxiliary, unlabeled non-sensitive and publicly available data. The student model is trained on this combination to accurately mimic the ensemble. Doing this ensures that the student does not depend on a single sensitive training data point and protects the privacy of training data [10]. Strong privacy guarantees are established by restricting the student training to a fixed and limited number of teacher votes and by exposing only the topmost vote after adding random noise generated from the *Laplacian Distribution*. The exposure of student to the teacher's knowledge needs to be quantified and bounded. Such approaches are generally applicable to machine learning algorithms with high utility and promises meaningful privacy. This technique establishes a precise privacy guarantee of the training data in an intuitive and rigorous manner.

Fig. 4. Private Aggregation of Teacher Ensemble



*2) Training Procedure:* During the data partitioning phase, it is ensured that the number of disjoint subsets is not too large with respect to the dataset size and task complexity. The teacher ensemble is deployed to make predictions on unseen inputs and these predictions are aggregated which accounts for the privacy guarantees of the teacher ensemble. Adding *Laplacian* noise introduces ambiguity and increases the privacy at the cost of accuracy of the predicted labels.

The privacy guarantee of the teacher ensemble stems from its aggregation. If there are 'm' number of classes in our task, the label count for a given class is $j \in |m|$ and the input $\vec{x}$ is the number of teachers that assigned class j to the input $\vec{x} : n_j(\vec{x}) =\mid i : i \in |n|, f_i(\vec{x}) = j \mid$.

Applying plurality, using the label with the largest count - the ensemble's decision will depend on a single teacher's vote. There is a possibility of a tie when the two labels have the vote count differing by at most one. In such cases, the aggregated output changes if one teacher makes a different prediction. The random noise that we add to the vote counts $n_j$ to introduce ambiguity can be mathematically represented as:

$$f(x) = argmax_j\{n_j(\vec{x}) + Lap(\frac{1}{\gamma})\} \quad (1)$$

In the above equation, $\gamma$ is the privacy parameter and Lap(b) is the Laplacian distribution with location 0 and scale b. Higher the value of $\gamma$, stronger is the privacy guarantee. The smooth sensitivity of the moments are bounded and the noise is proportionally added to the moments themselves through the beta parameter [20].

The privacy loss, in general, is determined using the number of queries made to the underlying model. As claimed, the privacy loss of the PATE approach is relatively lower than the other ones because of the fact that the student model is the one that is deployed in lieu of teacher ensemble. The privacy loss is independent of the number of queries made to the student model. In PATE, the privacy loss is attributed towards the number of queries made by the training student to the teacher ensemble. The user's privacy who contributed to the original training data set is preserved even if the student's architecture and internal parameters are made public or reverse-engineered by an adversary.

## V. Evaluation and Results

The two models discussed above have been implemented using Neural Networks. In the original papers, MNIST dataset has been tested by both the approaches. DP-SGD mentions the evaluation with CIFAR-10 dataset but does not provide sufficient testing results. In this section, we test both the MNIST and CIFAR-10 datasets with these two approaches and present the results. In the following sections we present the results obtained for the both datasets followed by a comparison of the two models.

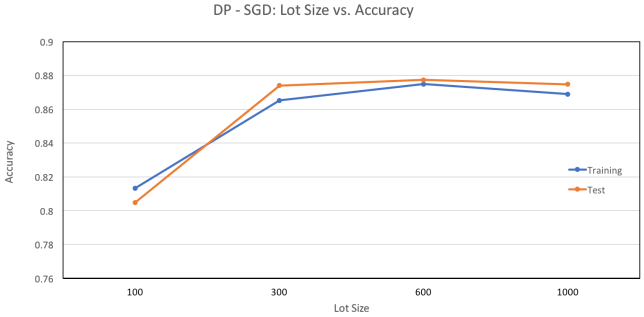
### A. Approach I

*1) MNIST:* Along with testing all the hyper-parameters extensively, Abadi et al. focused on a few important parameters which empirically affect the accuracy, privacy and over all computational activity of the model. It is observed that *lot size* (consists one or more batches which is different from the conventional batch size used in SGD), $\sigma$ and $\epsilon$ affect the privacy loss and accuracy more than any of the other hyper parameters. To test various hyper parameters, the default settings used in reference to Abadi et al. can be found in the Appendix Table VII.

*a) Principal Component Analysis:* In terms of the architecture, Abadi et al. claimed that applying *Principal Component Analysis (PCA)* with 60 dimensions, 1 hidden layer and 1000 hidden units gives better results than using 2 hidden layers and also increases the accuracy by almost 2%. We verify the claims by running the algorithm on MNIST and the results are shown in the TableIX. Our results verified the claim made by the authors and the results show that running time without PCA is almost 8 times more than compared to running it with PCA. Further, we tested the model with various other hyper parameters as listed in Appendix Table XIV. Hyper parameter search on the MNIST dataset was carried out extensively. As expected, lot size, sigma and epsilon did have the greatest impact on the accuracy and privacy loss.

*b) Lot Size:* The effect of the lot is the same as that of a batch. According to Abadi et al. [8], the lot size is computed such that $\frac{N}{L}$, where $N$ = Number of training images, $L$ = Lot size, remains within the privacy bound [8]. With a smaller lot size, the additive effect of the noise is greater as compared to the larger lot size which has relatively lesser effect on reducing the accuracy and is shown in Figure 5. Based on these results, we conclude that the optimal *lot size* is 600. This is the same as the results obtained by Abadi et al. Further, we look into the privacy loss as it is also an important factor and study the relationship further in the following sections.

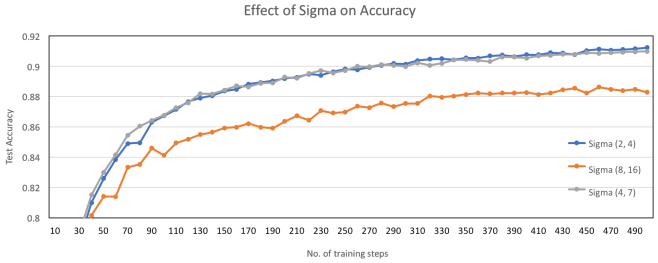
Fig. 5. Lot Size vs. Accuracy for MNIST Dataset




*c) Sigma($\sigma$):* Similar to the lot size, $\sigma$ is claimed to have a significant impact on the accuracy and privacy loss of the model [8]. Sigma determines the amount of noise being added during the gradient update step. There are two sigma parameters - one $\sigma$ value is used by the SGD method while the other, $\sigma_{pca}$ is used to add noise in PCA. Using the values presented by Abadi et al. [8] we tested 3 levels of noise, i.e. small $(2, 4)$, medium $(4, 7)$ and large (8, 16) and the results

are shown in figure 6. The results show that the accuracy does not vary much for the small and medium noise values and the average accuracy with medium noise is slightly better than noise value of 2. So, the optimal sigma value was found to be medium noise (4, 7).

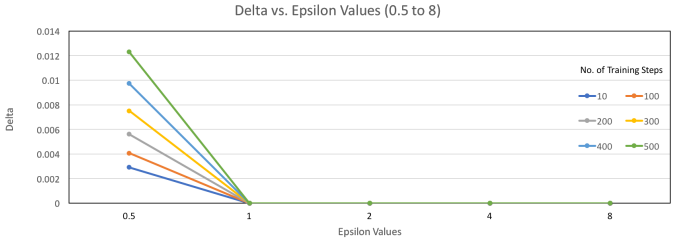
Fig. 6. Effect of Sigma ($\sigma$, $\sigma_{pca}$) on Accuracy for MNIST Dataset




*d) Epsilon ($\epsilon$):* As discussed earlier, the value of $\epsilon$ decides the amount of privacy of the data. It can be explained as a $logs$ ratio of the probability of getting data from $D$ over $D'$. If the $\epsilon$ value is too large, the adversary's chances of predicting the original source of data also increase. $\epsilon$ directly affects the value of the privacy achieved on the data. $\delta$ value is the probability with which the privacy model can be broken that it acts as a bound on privacy value. According to our implementation, we target the delta bound of $\delta = 10^{-5}$ which is consistent with the value used in original paper.

7 different epsilon values were tested and the results have been presented in the Figure 14. From this figure, we can speculate that the lowest $\delta$ values are obtained for $\epsilon$ = 0.5. Hence in Figure 7 below, we present the $\delta$ values for the range of $\epsilon$ values (0.5 - 8) and indeed, $\epsilon$ = 0.5 gives us the smallest $\delta$ values while maintaining the $\delta$ bound.

Fig. 7. MNIST: Delta Values vs. Epsilon Values (0.5 to 8)




*e) Number of Training Steps:* We also tested the number of training steps to see what effect it would have on the accuracy or the privacy loss. These results have been presented in the Appendix in Figures 13, 19 and Table X. The accuracy and $\delta$ keep increasing with number of epochs, however, we want to keep the value of $\delta$ as reasonably low as possible. So we concluded that training steps 500 are enough to give a high accuracy and a fairly low $\delta$ value which helps achieve better privacy.

These hyper parameter values give us a Testing accuracy of 90%. Refer to Table XVI for the values.

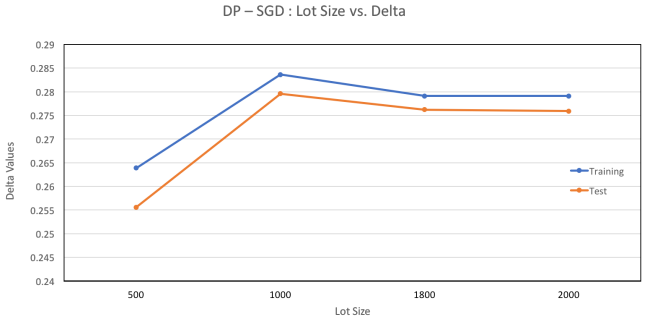
*2) CIFAR10:* Based on the results observed for MNIST, our hyper parameter search was limited to testing fewer values for each parameter. We focused on finding the optimal values for each parameter based on observations made from the MNIST dataset.

*a) Lot Size:* We tested 4 lot sizes and results can be seen in Table I and Figure 8.From these results we can conclude that lot size = 1000 is optimal for CIFAR-10 dataset.

TABLE I
CIFAR WITH DP-SGD: LOT SIZE VS. DELTA VALUES WITH $\sigma$=(4,7), $\epsilon$=1.0

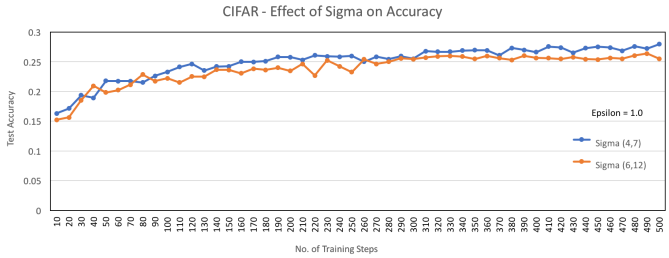
| Lot Size | Delta Values |
|---|---|
| 500 | 1.95e-08 |
| 1000 | 3.84e-05 |
| 1800 | 0.014262491 |
| 2000 | 0.030357025 |

Fig. 8. Lot Size vs. Accuracy for CIFAR Dataset



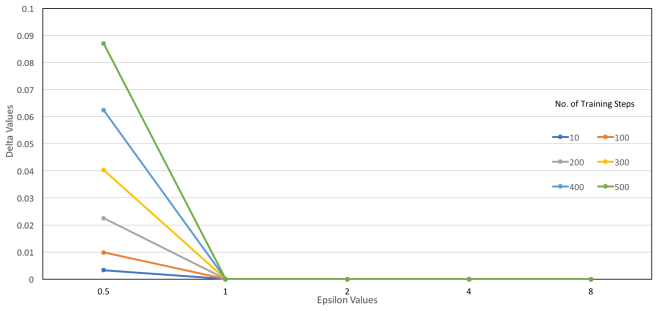

*b) Sigma:* We tested two sigma values - medium noise (4, 7) and large noise (6, 12). Here, we observe that adding medium noise gave us optimal results. A smaller noise would have given us better accuracy but it would have reduced the privacy. This can be seen in Figure 9.

Fig. 9. CIFAR: Effect of Sigma ($\sigma$, $\sigma_{pca}$) on Accuracy - All Training Steps



*c) Epsilon:* We tested 7 epsilon values with the CIFAR dataset as well. We observed that for epsilon value 0.5 we maintain delta values well above the bound of $10^{-05}$. This can be seen in Figure 10 and Figure 15 in appendix.

Fig. 10. CIFAR: Delta Values vs. Epsilon Values (0.5 to 8)




With CIFAR, we did not test the number of training steps because with MNIST we observed that increasing the training steps increases the $\delta$ value while not increasing the accuracy as much. Hence we used 500 training steps for CIFAR10 as well. These hyper parameter values give us a Testing accuracy of 28%. Refer to Table XVII for the values.

For both sets, we used the clipping norm proposed by Abadi et al [8]. It should be noted that changing the clipping norm also has an effect on the results of differential privacy. Since it was extensively studied by Abadi et al. we decided to stick to the default values presented in Appendix Table VII.

## B. Approach II

Papernot et al. emphasize that the number of teachers trained in teachers model and the teacher id used to train the students greatly impact the results obtained. Therefore, we decided to investigate the results by tempering these parameters and have presented the optimal results obtained for teachers ensemble selected. The default values used by the original paper are shown in Table VIII and the hyper parameter values used in experiment are shown in Table XV.

*1) MNIST:*

*a) Teacher ID:* The students model is trained with a particular teacher id as discussed in the implementation section above. The teacher id selected is the one with the optimal results in teacher ensemble. In order to select the best teacher id, we run the algorithm for each teacher id. Hence, for an ensemble of 250 teachers we must train and test each teacher with 0-249 instances. Due to time limitation, we assumed the teacher id 10 as presented in Papernot et al. to be the optimal 250 (number of teachers) teacher ensemble result. In order to compute results for 10 and 100 teachers, we used arbitrary teacher values of 4 and 10 respectively.

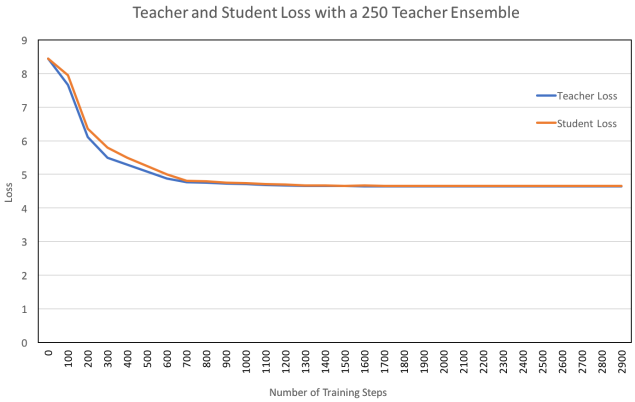
*b) Number of Teachers to Train:* Referencing the values tested by Papernot et al., we tested the results for training with 10, 100 and 250 teachers. The results are present in Table II below.

It is evident that training with 250 teachers gives us the highest accuracy. To understand the impact of the number of teachers on the loss, we plot the loss versus number of teachers shown in Figures 16, 17 in the appendix and 11. We observe that With 250 teachers, we maintain a more consistent teacher and student loss, even though the loss is a bit higher than 100 teachers.

TABLE II
MNIST: NUMBER OF TEACHERS VS TEACHER ACCURACY AND STUDENT ACCURACY

| Number of Teachers | Teacher Accuracy | Student Accuracy |
|---|---|---|
| 10 | 0.9778 | 0.3168 |
| 100 | 0.8535 | 0.8694 |
| 250 | 0.8584 | 0.8938 |

Fig. 11. 250 Teacher Ensemble - Loss




*c) Number of Training steps:* From Figure 11, we also noticed that running the ensemble for 3000 steps (the set default) was actually unnecessary. The loss becomes constant on reaching 1300 steps. Hence, to obtain optimal results, we only computed the teacher ensemble for 1300 steps and therefore trained the student for 1300 steps as well.

*d) Privacy Analysis for MNIST:* There were a few parameters in the privacy analysis that we could alter to reduce our privacy cost.The hyper parameters and the results of altering their values are described below:

The $\beta$ **parameter**, as discussed in the implementation section, is added to ensure that the final $\epsilon$ value isn't data dependent and is not easily revealed. So when we change this parameter, the actual $\epsilon$ value of the model isn't changed. However, the target differential privacy value is altered. We want to ensure that the two $\epsilon$ values are as close as possible because if they are very far apart it means we need to add more noise. This is quite costly in terms of privacy preservation. The effects of changing the $\beta$ values are presented in Table III below:

The **Noise** $\epsilon$ **parameter** changes the amount of noise being added to the teacher's vote count as stated in equation (1) in Section IV.d. Making this value too large could affect the accuracy of the label prediction on cost of achieving more privacy.

TABLE III
MNIST: $\beta$ VERSUS PRIVACY COST

| $\beta$ | DP-$\epsilon$ value | Actual $\epsilon$ value | Noise |
|---|---|---|---|
| 0.09 | 2.07232658369 | 1.47052484318 | 0.965098848674 |
| 0.15 | 3.45387763949 | 1.47052484318 | 0.579059309204 |
| 0.01 | 0.230258509299 | 1.47052484318 | 8.68588963807 |
| 0.05 | 1.1512925465 | 1.47052484318 | 1.73717792761 |
| 0.07 | 1.6118095651 | 1.47052484318 | 1.24084137687 |
| 0.06 | 1.3815510558 | 1.47052484318 | 1.44764827301 |

TABLE IV
MNIST: NOISY $\epsilon$ VERSUS PRIVACY COST

| Noisy $\epsilon$ | DP-$\epsilon$ value | Actual $\epsilon$ value | Noise |
|---|---|---|---|
| 0.01 | 1.6118095651 | 1.43911628681 | 1.24084137687 |
| 0.05 | 1.6118095651 | 1.47052484318 | 1.24084137687 |
| 0.07 | 1.6118095651 | 1.43956559567 | 1.24084137687 |

Based on these two tables we conclude that the values of $\beta$ = 0.07 and noisy $\epsilon$ = 0.05 obtain optimal results.

These hyper parameter values give us a Teacher Training accuracy of 85% and Student Training accuracy of 89%. Refer to Table XVIII for the values.

*2) CIFAR10:*

*a) Number of Teachers:* Doing similar tests as that of MNIST we found that number of teachers = 100 gives us better results than number of teachers = 250. Hence for this implementation we used number of teachers = 100.

*b) Teacher ID:* We paid a lot of attention to the Teacher ID for this section because of how important it is in terms of the results. Here we decided to run a few tests with different teacher IDs and choose the ensemble that yielded the best result. We chose 10 numbers randomly and computed and saved teacher ensembles accordingly. It is very important to save the ensemble because the disjoint subsets are created randomly every time we run the algorithm. Hence if with teacher_id = 23 you get the best results and you don't save it, the next you run the algorithm with the same ID it may not be the same. The teacher training accuracy for different teacher ids has been presented in Table XIII in the Appendix.

Based on this table we concluded that Teacher ID 23 gave the best teacher training accuracy of 36.3%.

*c) Number of Training Steps:* As we can see in Figure 12 below after around 1500 steps the loss value begins to plateau. This was the case for all IDs and hence the final result was only computer for 1500 steps. The figure below provides the result for only teacher ID 71.

*d) Privacy Analysis:* Similarly for CIFAR-10, we analyzed the privacy cost by altering the values of $\beta$ and Noisy $\epsilon$. These results have been presented in the Tables V and VI.

Based on the results obtained in these tables we conclude that $\beta$ = 0.09 and noisy $\epsilon$ = 0.1.

These hyper parameter values give us a Teacher Training accuracy of 36.3% Student Training accuracy of 33%. Refer to Table XIX for the values.

Fig. 12. CIFAR: No. of Training Steps vs. Privacy Loss for Teacher ID = 71

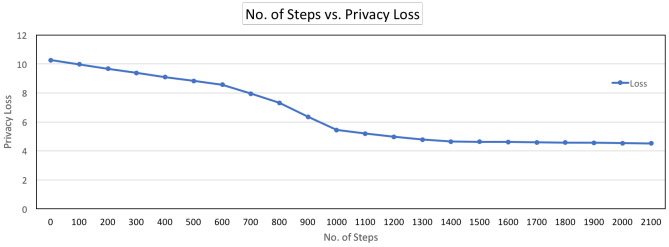



TABLE V
CIFAR: NOISY $\epsilon$ VERSUS PRIVACY COST

| Noisy $\epsilon$ | DP-$\epsilon$ value | Actual $\epsilon$ value | Noise |
|---|---|---|---|
| 0.15 | 3.45387763949 | 2.3134801809 | 0.579059309204 |
| 0.07 | 1.6118095651 | 2.3134801809 | 1.24084137687 |
| 0.09 | 2.07232658369 | 2.3134801809 | 0.965098848674 |

## VI. DISCUSSION

Based on the results obtained, we observe that the accuracy obtained by DP-SGD is not even close to what is reported in the original paper except when it is run with 1000 iterations which yields an accuracy of 91.1%. With 500 iterations, our optimal result was an accuracy of 90.85%. However, our results concord with all the other claims made. Applying PCA significantly reduced the time required to train the model and also improved the accuracy without the addition of extra hidden layers. The main goal of this algorithm was to achieve optimal accuracy combined with minimalistic privacy loss. Since, applying differential privacy always affect the accuracy of predictions, the lower privacy loss of data justifies the drop in accuracy for DP-SGD. However, it does not justify the difference of almost 10% in the accuracy results compared to the accuracy reported in the paper. Introducing the concept of lot size did help in maintaining the variance over batches used in SGD. Running DP-SGD on CIFAR-10 did not achieve good accuracy. The average accuracy achieved was around 28% and required almost 10 times more iterations to train the model as compared to MNIST. Given the complex nature of CIFAR-10 dataset, it would have achieved better accuracy if ran with Deep-CNN implemented with some preprocessing techniques like cropping, ZCA whitening, SIFT, SURF etc.

DP-Multiple teachers technique obtained a teacher's accuracy of 85% and highest accuracy of 89% with students model on MNIST dataset trained with 250 teachers. With CIFAR10 where the teacher's accuracy obtained is 36.5% while student's accuracy 33.6%. DP-MT technique randomly partitions the data into numerous datasets based on the number of teachers, which is why we cannot compare our results with the results reported in paper. Due to the randomness, which makes it more robust to the model-inversion attacks, the accuracy obtained

TABLE VI
NOISY EPSILON

| Noisy $\epsilon$ | DP-$\epsilon$ value | Actual $\epsilon$ value | Noise |
|---|---|---|---|
| 0.05 | 2.07232658369 | 20.7564627325 | 0.965098848674 |
| 0.1 | 2.07232658369 | 2.3134801809 | 0.965098848674 |
| 0.15 | 2.07232658369 | 1.45786781952 | 0.965098848674 |

with teacher id 10 upon training with 250 teachers was not similar to the one obtained in the original paper. DP-Multiple teachers technique obtained a similar accuracy as DP-SGD on MNIST dataset but performed better on CIFAR-10 dataset. One of the reasons for DP-MT to obtain better accuracy on CIFAR-10 is the use of Deep CNN model. Although, we cannot directly compare the performance of these two approaches in terms of the privacy preservation because of two completely different underlying techniques, but given a dataset like MNIST DP-Multiple Teachers performs better in terms of privacy preservation.

In conclusion, these techniques are well suited for datasets like MNIST, i.e. gray scale images with minimal background noise. They do not generalize well on different datasets and there is a lot of scope for future work with these two approaches. Given more time, we would have tried implementing the DP-SGD with CNN in order to have a good comparison between the two approaches. At last, we conclude that given a dataset similar to MNIST, DP-SGD performed better in terms of accuracy but on the cost of higher processing time whereas DP-MT achieved a slightly less accuracy than DP-SGD with substantially lesser training and testing time.

## VII. RELATED WORK

There are several privacy protection techniques that are used in machine learning models. They range from being simple fixes like anonymization to more complex changes like manipulating the state-of-art machine learning models. The general misconception is that to protect private information, we can always anonymize a "rich" dataset containing sensitive information [10]. This is an easy fix and intuitively, hiding several aspects of data will definitely preserve the privacy. However, this richness allows for the identification of an individual through a combination of quasi-fields like "zip code" or "date of birth" of a person [10]. Hence, on using this anonymized dataset with a non-anonymized dataset from another source, the adversary can still gain access to sensitive information. We want our models to prevent such "linkage attacks" [10]. Differential privacy addresses this very issue. The method is developed such that it is learning nothing about the individual whilst learning the population [10]. So, with differential privacy, the presence or absence of auxiliary information, something that an adversary may have access to, will not provide any information about the individual [10]. It is able to do so by introducing randomness to the entire dataset, thereby creating a new similar dataset but with completely new records. Laplace Mechanism is one such way to add noise. The true value is computed and some noise from the Laplace Distribution is added[2]. When implemented with a model that can use approximate statistics to make a prediction, it generates good results. However, privacy loss is added for each query made. So, if there are many queries being made, then the database needs to be large enough in order to maintain accuracy and privacy.

Perturbation models are good in preserving privacy, however, they reduce the accuracy of data being released. With anonymity, there is no impact on the accuracy of the model. An anonymity model that has been extensively researched is k-anonymity. It aims to suppress or generalize attributes until each row in the database is identical with at least k-1 other rows[3]. Hence, in the worst case the adversary can also create a linkage to an entry in a pool of k individuals. However, the method is still subject to attacks such as: unsorted matching attack; complementary release attack; temporal attack [11].

Many perturbation models have been studied in the recent past. Stochastic Gradient Descent is one such model that has proved to be significantly important for differential privacy because it is a simple model which satisfies the same asymptotic guarantees as more computationally intensive learning methods [12]. For low-dimensional data, SGD with mini batches tend to maintain the same accuracy with privacy preservation as compared to accuracy of vanilla model [12]. However, increase in dimensionality increases the number of features which tends to reduce the generalization accuracy. There are many methods available to reduce dimensionality like k-SVM and Principal Component Analysis (PCA). PCA is a matrix factorization method which learns the linear projection of a dataset into a lower dimension while maintaining as much variance as possible from the original dataset [13].

Differentially private SGD iteratively updates the weights by adding some noise drawn from a distribution like the Laplacian Distribution discussed above [12]. This is known as an output perturbation method [13]. However, there are other models which follow a different approach. For example, there are algorithms that use a sampling and aggregation framework [13]. Here, the sample space is divided into many subsets. The results from each subset are combined and noise is added to the aggregated subset [13].

Most of the methods in output perturbation models need to be bounded [13]. The sampling and aggregation framework doesn't have this limitation however, they are often implemented with a bound [13].

Machine learning tries to invoke generalization by training a subset of samples resulting in models that are quite robust and are able to avoid the effects of randomness. Privacy models just add some kind of randomness. So a model that performs better with privacy proves that that model is more robust. We could say that both privacy and machine learning, in their own way, have the same goal: improve the generalization capabilities of a model [13].

[2]http://www.cs.cmu.edu/~avrim/ML07/learning-and-privacy.pdf
[3]https://www.cs.cmu.edu/~jblocki/Slides/K-Anonymity.pdf

## VIII. Statement of Contributions

The work was divided equally among all the team members. Inderjot and Nirmal started by studying about the two approaches, whereas Taanvi ran the algorithms to obtain the results used for comparison. We contributed equally in writing the report, preparation of the presentation and to the insights presented on the results. We hereby state that all the work presented in this report is that of the Authors.

## References


[1] CIFAR-10 and CIFAR-100 datasets. www.cs.toronto.edu/~kriz/cifar.html.

[2] TensorFlow convolutional neural networks tutorial. www.tensorflow.org/tutorials/deepcnn.

[3] TensorFlow Models - Differential Privacy. https://github.com/tensorflow/models/tree/master/differential_privacy.

[4] Peterson, Andrea. *"Why a staggering number of Americans have stopped using the Internet the way they used to."* Washington Post. May 13 (2016).

[5] Green, Matthew, and Johns Hopkins. *"A few thoughts on cryptographic engineering."* (2013).

[6] Millier, Samantha L. *"Facebook Frontier: Responding to the Changing Face of Privacy on the Internet, The."* Ky. LJ 97 (2008): 541.

[7] M. Fredrikson, S. Jha, and T. Ristenpart. *"Model inversion attacks that exploit confidence information and basic countermeasures"*. In CCS, pages 1322-1333. ACM, 2015.

[8] Abadi, Martn, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. *"Deep learning with differential privacy."* In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308-318. ACM, 2016.

[9] Papernot, Nicolas, Martn Abadi, lfar Erlingsson, Ian Goodfellow, and Kunal Talwar. *"Semi-supervised knowledge transfer for deep learning from private training data."* arXiv preprint arXiv:1610.05755 (2016).

[10] Cynthia Dwork and Aaron Roth. *"The Algorithmic Foundations of Differential Privacy"* Foundations and Trends in Theoretical Computer Science Vol. 9, Nos. 34 (2014) 211407

[11] LATANYA SWEENEY *"k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY1"* (2002)

[12] Shuang Song, Kamalika Chaudhuri and Anand D. Sarwate *"Stochastic gradient descent with differentially private updates"* (2014)

[13] Zhanglong Ji, Zachary C. Lipton and Charles Elkan *"Differential Privacy and Machine Learning: a Survey and Review"* (2014)

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haner. *"Gradient-based learning applied to document recognition"*. Proceedings of the IEEE, 86(11), 1998.

[15] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang. *"Analyze Gauss: Optimal bounds for privacy-preserving principal component analysis"*. In STOC, pages 11-20. ACM, 2014.

[16] C. Dwork, G. N. Rothblum, and S. Vadhan. *"Boosting and dierential privacy"*. In FOCS, pages 51-60. IEEE, 2010.

[17] F. D. McSherry. *"Privacy integrated queries: An extensible platform for privacy-preserving data analysis"*. In SIGMOD, pages 19-30. ACM, 2009.

[18] Dietterich, Thomas G. *"Ensemble Methods in Machine Learning"*. Multiple Classifier Systems: First International Workshop, Proceedings, pages 19-30. ACM, 2009.

[19] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. *"Our data, ourselves: Privacy via distributed noise generation."* In EUROCRYPT, pages 486-503. Springer, 2006.

[20] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith *"Smooth sensitivity and sampling in private data analysis"* In Proceedings of the 39th annual ACM Symposium on Theory of Computing, pp. 7584. ACM, 2007


## Appendix

TABLE VII
Default Values used for Hyper-parameter evaluation in DP-SGD

| Hyper-parameter | Values |
|---|---|
| PCA Dimension | 60 |
| No. of Hidden Layers | 1 |
| No. of Hidden Units | 1000 |
| Sigma | (4,7) |
| Lot Size | 600 |
| No. of Training Steps | 500 |

TABLE VIII
Default Values used for Hyper-parameter evaluation in DP-Multiple Teachers

| Hyper-parameter | Values |
|---|---|
| No. of Teachers | 250 |
| Training epochs | 3000 |
| Teacher Id | 10 |

Fig. 13. MNIST: Training Steps vs. Accuracy (DP-SGD)

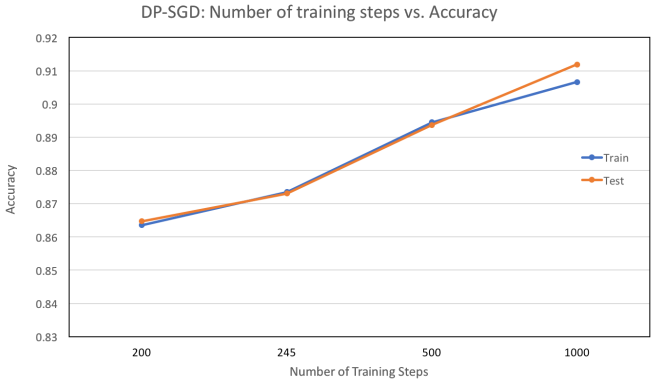



Fig. 14. MNIST: Delta Values vs. Epsilon Values (0.125 to 8)

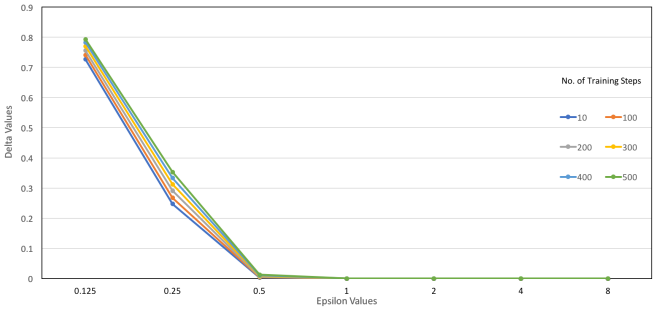


TABLE IX
DP-SGD WITH/WITHOUT PCA WITH MNIST DATASET

| PCA projections | Np. of Hidden Layers | Sigma | PCA sigma | Training Accuracy(%) | Test Accuracy(%) | Running time(s) |
|---|---|---|---|---|---|---|
| 0 | 1 | 4 | 7 | 86.52 | 86.84 | 3467 |
| 0 | 2 | 4 | 7 | 86.12 | 85.92 | 8373 |
| 60 | 1 | 4 | 7 | 87.49 | 87.73 | 1434 |

Fig. 15. CIFAR: Delta Values vs. Epsilon Values (0.125 to 8)

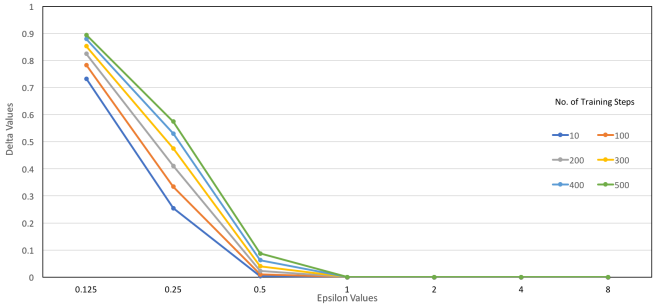



Fig. 16. 10 Teacher Ensemble - Loss

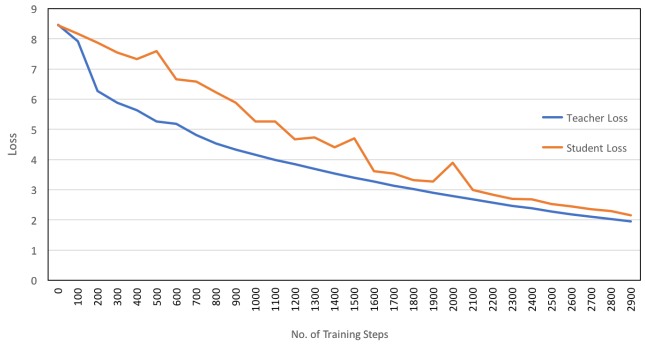



Fig. 17. 100 Teacher Ensemble - Loss

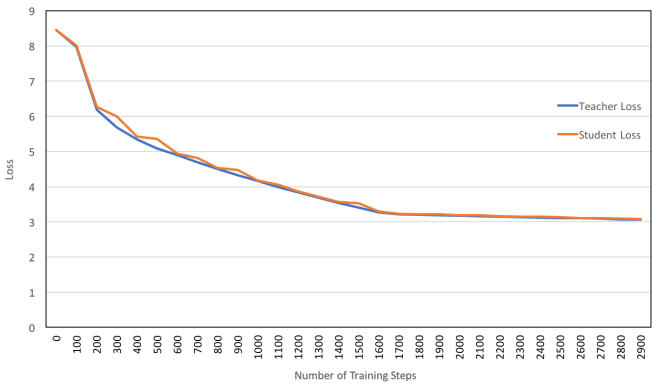



Fig. 18. No. of training steps vs. Delta Values (DP-SGD)

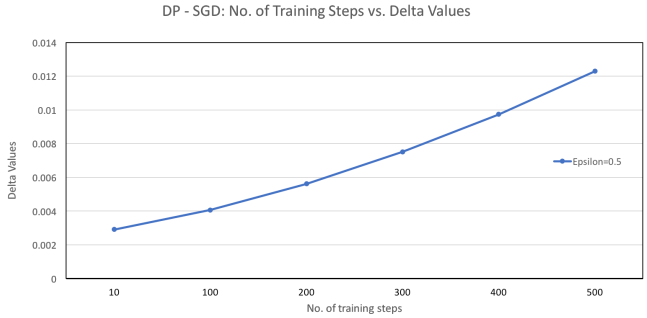



Fig. 19. No. of training steps vs. Accuracy (Epsilon = 0.5)

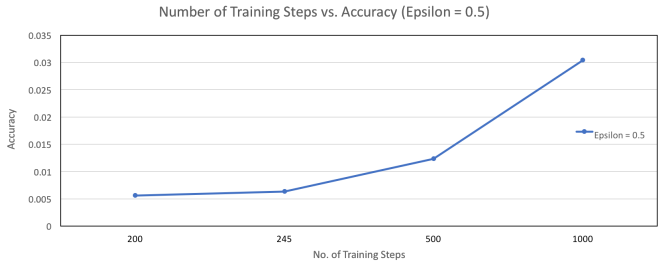



TABLE X
DP-SGD: NO. OF TRAINING STEPS VS ACCURACY AND PRIVACY WITH TARGET DELTA $(10^{-5})$

| Number of Training Steps | Test Accuracy(%) | Epsilon | Delta |
|---|---|---|---|
| 200 | 86.47 | 0.5 | 5.6e-03 |
| 245 | 87.31 | 0.5 | 6.3e-03 |
| 500 | 89.37 | 1.0 | 1.9e-07 |
| 1000 | 91.19 | 1.0 | 5.4e-07 |

TABLE XI
DP-MT: PARAMETERS TESTED

| Noise Epsilon |
|---|
| Beta |

TABLE XII
MNIST WITH DP-SGD: LOT SIZE VS. DELTA VALUES WITH $\sigma$=(4,7), $\epsilon$=0.5

| Lot Size | Delta Values |
|---|---|
| 100 | 2.8e-03 |
| 300 | 3.5e-03 |
| 600 | 6.3e-03 |
| 1000 | 1.7e-02 |

TABLE XIII
TEACHER_IDS AND THE CORRESPONDING PRECISIONS

| Teacher ID | Precision |
|---|---|
| 0 | 0.3174 |
| 3 | 0.3278 |
| 7 | 0.3502 |
| 18 | 0.3342 |
| 23 | 0.3636 |
| 25 | 0.3517 |
| 49 | 0.2921 |
| 50 | 0.3146 |
| 71 | 0.3301 |
| 99 | 0.3444 |

TABLE XIV
DIFFERENTIAL PRIVACY - STOCHASTIC GRADIENT DESCENT

| Hyper-Parameters | MNIST | CIFAR |
|---|---|---|
| Lot Size | 100 | 1000 |
| | 300 | 1800 |
| | 600 | 2000 |
| | 1000 | ... |
| No. of Training Steps | 200 | 500 |
| | 245 | ... |
| | 500 | ... |
| | 1000 | ... |
| Epsilon | 0.125 | 0.125 |
| | 0.25 | 0.25 |
| | 0.5 | 0.5 |
| | 1 | 1 |
| | 2 | 2 |
| | 4 | 4 |
| | 8 | 8 |
| Sigma $(\sigma, \sigma_{pca})$ | (2,4) | (4,7) |
| | (4,7) | (6,12) |
| | (8,16) | ... |
| Delta[4] | $10^{-5}$ | $10^{-5}$ |

TABLE XV
MULTIPLE TEACHERS - TEACHER ENSEMBLE

| Hyper-Parameters | MNIST | CIFAR |
|---|---|---|
| Number of Teachers | 10 | 100 |
| | 100 | ... |
| | 250 | ... |
| | 1000 | ... |
| Number of Training Steps | 3000 | 2200 |
| | 1500 | 1500 |
| Number of Epochs / Delay | 200 | 350 |
| | 350 | ... |
| | 500 | ... |
| Teacher ID | 10 [9] | Tested Many |

TABLE XVI
OPTIMAL HYPER PARAMETER VALUES: MNIST-DP-SGD

| Parameter | Value |
|---|---|
| Lot Size | 600 |
| No. of Training Steps | 500 |
| Delta | 1e-05 |
| Sigma | (4, 7) |
| Epsilon | 0.5 |

TABLE XVII
OPTIMAL HYPER PARAMETER VALUES: CIFAR10-DP-SGD

| Parameter | Value |
|---|---|
| Lot Size | 1000 |
| No. of Training Steps | 500 |
| Delta | 1e-05 |
| Sigma | (4, 7) |
| Epsilon | 0.5 |

TABLE XVIII
OPTIMAL HYPER PARAMETER VALUES: MNIST-DP-MULTIPLE TEACHERS

| Parameter | Value |
|---|---|
| No. of Teachers | 250 |
| Teacher ID | 10 |
| No. of Training Steps | 1300 |
| $\Delta$ | 1e-05 |
| $\beta$ | 0.07 |
| Noisy Epsilon | 0.05 |

TABLE XIX
OPTIMAL HYPER PARAMETER VALUES: CIFAR10-DP-MULTIPLE TEACHERS

| Parameter | Value |
|---|---|
| No. of Teachers | 100 |
| Teacher ID | 23 |
| No. of Training Steps | 1500 |
| Delta | 1e-05 |
| beta | 0.09 |
| Noisy Epsilon | 0.1 |