Wireless Networks
https://doi.org/10.1007/s11276-022-03113-7

Can differential privacy practically protect collaborative deep learning inference for IoT?

Jihyeon Ryu[1] · Yifeng Zheng[2] · Yansong Gao[3] · Alsharif Abuadbba[4] · Junyaup Kim[1] · Dongho Won[1] · Surya Nepal[4] · Hyoungshick Kim[1] · Cong Wang[5]

Accepted: 18 August 2022

Abstract

Collaborative inference has recently emerged as an attractive framework for applying deep learning to Internet of Things (IoT) applications by splitting a DNN model into several subpart models among resource-constrained IoT devices and the cloud. However, the reconstruction attack was proposed recently to recover the original input image from intermediate outputs that can be collected from local models in collaborative inference. For addressing such privacy issues, a promising technique is to adopt differential privacy so that the intermediate outputs are protected with a small accuracy loss. In this paper, we provide the first systematic study to reveal insights regarding the effectiveness of differential privacy for collaborative inference against the reconstruction attack. We specifically explore the privacy-accuracy trade-offs for three collaborative inference models with four datasets (SVHN, GTSRB, STL-10, and CIFAR-10). Our experimental analysis demonstrates that differential privacy can practically be applied to collaborative inference when a dataset has small intra-class variations in appearance. With the (empirically) optimized privacy budget parameter in our study, the differential privacy technique incurs accuracy loss of 0.476%, 2.066%, 5.021%, and 12.454% on SVHN, GTSRB, STL-10, and CIFAR-10 datasets, respectively, while thwarting the reconstruction attack.

Keywords Collaborative inference · Differential privacy · Data reconstruction attack · Cloud computing

✉ Yifeng Zheng
yifeng.zheng@hit.edu.cn

Jihyeon Ryu
jhryu@security.re.kr

Yansong Gao
yansong.gao@njust.edu.cn

Alsharif Abuadbba
Sharif.Abuadbba@data61.csiro.au

Junyaup Kim
yaup22cc@likelion.org

Dongho Won
dhwon@security.re.kr

Surya Nepal
Surya.Nepal@data61.csiro.au

Hyoungshick Kim
hyoung@skku.edu

Cong Wang
congwang@cityu.edu.hk

[1] Department of Computer Science and Engineering, Sungkyunkwan University, Seoul, South Korea

[2] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

[3] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

[4] Data61, CSIRO, Sydney, Australia

[5] Department of Computer Science, City University of Hong Kong, Hong Kong, China

Published online: 05 September 2022

1 Introduction

Recent advancements in deep learning techniques have greatly empowered various Internet of Things (IoT) applications such as object recognition, human activity recognition, health monitoring, and environmental sensing [1–4]. However, running a trained deep neural network (DNN) model with new inputs (i.e., DNN inference) would be resource-intensive and requires massive computational resources, making it notably difficult to be directly deployed on resource-constrained IoT devices [5, 6]. Therefore, an alternative practical way for deployment is to construct a DNN model on a cloud server and forward input data from IoT devices to the cloud server for the inference. With such deployment, however, IoT devices' data are inevitably exposed to the cloud service provider, raising privacy concerns for some IoT applications that would process sensitive and/or private data.

Recently, *collaborative inference* [7, 8] was introduced to avoid the direct exposure of such data from resource-constrained IoT devices, which DNN inference can still be effectively supported. In particular, in the collaborative inference framework, a DNN model is split into a local part model containing simple shallow layers of the DNN model and a remote part model containing the remaining sophisticated layers. The local part model is typically deployed on the resource-constrained IoT devices, while the remote part model is deployed on the cloud, as illustrated in Fig. 1.

In this collaborative inference framework, DNN inference is performed collaboratively, crossing from the local part model to the remote part model. The local part model first processes input data to obtain an intermediate output. Then, this intermediate output is sent to the remote part model to perform forward inference computation over the remaining layers. Consequently, collaborative inference fundamentally eschews direct exposure of the raw input data to the cloud. Moreover, collaborative inference has clear advantages for reducing the computational resources of IoT devices in deep learning applications. In the view of model providers, the use of collaborative inference would be preferred as well because they do not need to give out the entire DNN model for deployment on local devices.

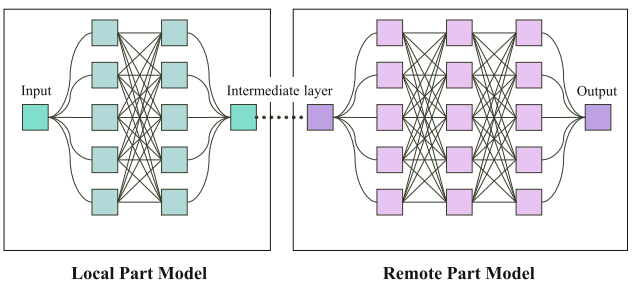


Fig. 1 Overview of collaborative inference

At first glance, it may seem sufficient to use collaborative inference for protecting raw input data used in a DNN model. However, recent studies [9, 10] show that collaborative inference could entail privacy risks. The intermediate output produced from the local part model can contain sensitive information used to recover the original raw input data. He *et al.* [10] presented the feasibility of a reconstruction attack targeting collaborative inference for image-based applications, which is designed for an honest-but-curious cloud service provider to recover the original input image from the intermediate output generated from the local part model. In independent work, to mitigate the privacy risks from exposing the intermediate output, Wang *et al.* [9] proposed a collaborative inference framework using *differential privacy* [11] to avoid the privacy leakage from the intermediate output. Differential privacy has become the de facto privacy standard as it provides a rigorous mathematical framework for formalizing privacy guarantees in terms of the privacy budget ϵ. The framework in [9] employs differential privacy via adding delicately calibrated noises to the intermediate output values. As such perturbations definitively incur a degradation on the inference accuracy, the framework further delicately provides a noisy training technique to endow the remote part model with robustness to perturbed data and alleviate the impact of noise perturbation on the inference accuracy.

To deploy a collaborative inference framework using differential privacy in the real world, it would be necessary for a given collaborative inference model to show that a reasonable budget ϵ for differential privacy can be chosen against such data reconstruction attacks. However, Wang *et al.* [9] did not thoroughly analyze the privacy-accuracy trade-offs in the presence of the state-of-the-art data reconstruction attack against collaborative inference. Therefore, our work was motivated by the following research question:

> **RQ:** Is it feasible to adopt differential privacy to gain protection in collaborative inference against reconstruction attack while preserving high accuracy of the inference?

To answer the research question, we implement the state-of-the-art differential privacy framework for collaborative inference [9], and newly apply the state-of-the-art reconstruction attack against that framework over various datasets to reveal the privacy-accuracy trade-offs. To our best knowledge, our study is the first that assesses the practical usability of differential privacy for collaborative inference in the presence of the state-of-the-art reconstruction attack. We summarize the key contributions of this paper as follows:

- We implement the state-of-the-art collaborative inference framework using differential privacy [9, 10] and reconstruction attack to analyze the privacy-accuracy trade-offs in collaborative inference.
- We conduct extensive evaluations on the attack and defense implementations with various datasets, including SVHN, GTSRB, STL-10, and CIFAR-10, and varying the privacy budget ϵ. Unlike the previous study with a fixed split [9], we evaluate several split settings by varying the layers of the local part model and the remote part model. We find that the effectiveness of differential privacy increases as the number of layers of the local part model decreases in the collaborative inference model deployment (i.e., the number of layers of the remote part model increases).
- We reveal insights about how the effectiveness of differential privacy is significantly affected by the characteristics of datasets through our experiments. We find that differential privacy would be more effective when a dataset has small intra-class variations in appearance. In our experiments, the best privacy budget ϵ incurs accuracy loss of 0.476%, 2.066%, 5.021%, and 12.454% on SVHN, GTSRB, STL-10, and CIFAR-10 datasets, respectively, while preventing data reconstruction attacks.

The remainder of this paper is organized as follows. Section 2 provides background information on the differential privacy-based collaborative inference framework and the data reconstruction attack. Section 3 presents comprehensive experimental evaluations. Section 4 discusses key findings from our extensive evaluations and draws practical insights. Section 5 describes the related work. Section 6 concludes this work.

2 Background

2.1 Collaborative inference

In the collaborative inference framework [7, 8] for IoT-cloud applications, as shown in Fig. 1, a trained DNN model, denoted by f_θ and parameterized by model parameters θ, is split into two parts: a local part model f_{θ_1} and a remote part model f_{θ_2}. The former is deployed on the client-side (resource-limited IoT devices), while the latter is on the cloud side. To perform inference for a data sample $\mathbf{x}$, the client first feeds $\mathbf{x}$ to the local part model and obtains $\mathbf{x}^* = f_{\theta_1}(\mathbf{x})$, which represents an intermediate output. This intermediate output is then sent to the cloud, which further applies the remote part model f_{θ_2} to $\mathbf{x}^*$ and produces the ultimate inference result $y = f_{\theta_2}(\mathbf{x}^*)$.

2.2 Differential privacy

Differential privacy is a mathematical framework defined for privacy-preserving data analysis. The formal definition of ϵ-differential privacy is as follows [11].

Definition 1 Given any two neighboring inputs D and D' which differ in only one data item, a mechanism $\mathcal{M}$ provides ϵ-differential privacy if $Pr[\mathcal{M}(D) \in S] \leq e^\epsilon \cdot Pr[\mathcal{M}(D') \in S]$.

Intuitively, the above definition indicates that for any output in the range S of the mechanism $\mathcal{M}$, its probability of being produced from D is very close to that of being produced from D', as characterized by ϵ. That is, given any output, one can hardly tell whether it is produced from D or D'. The parameter ϵ is usually referred to as the privacy budget. A smaller ϵ value indicates *stronger* privacy protection.

To achieve differential privacy, the common approach is to add calibrated noises to the output of a function $g(\cdot)$ based on specific probability distributions [12]. A widely used probability distribution in differential privacy is the Laplace distribution, denoted by $Lap(b)$, where b is called the scale parameter. In particular, the probability density function is: $Pr[x] = \frac{1}{2b} e^{-|x|/b}$. The Laplace mechanism [12] for ϵ-differential privacy works by sampling noises from $Lap(b)$ and adding the noises to the output values of the function $g(\cdot)$. Here, to achieve ϵ-differential privacy, b is set according to the global sensitivity Δg of the function $g(\cdot)$, i.e., $b = \frac{\Delta g}{\epsilon}$. Let $||\cdot||_1$ denote the l_1 norm. The global sensitivity of Δg is defined as:

$$\Delta g = \max_{D,D'} ||g(D) - g(D')||_1.$$

Algorithm 1 Differential Privacy Scheme Using Local Perturbation

Input: Input data sample $\mathbf{x}$; Bound threshold B; Privacy budget ϵ.
Output: Noisy intermediate output $\mathbf{x}^*$.
1: $\tilde{\mathbf{x}} \leftarrow f_{\theta_1}(\mathbf{x})$
2: $d = ||\tilde{\mathbf{x}}||_\infty$
3: $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} / max(1, \frac{d}{B})$
4: $\mathbf{x}^* \leftarrow \tilde{\mathbf{x}} + Lap(2B/\epsilon)$ // Sampling and adding noises element-wise.

2.3 Differential privacy for collaborative inference

The differential privacy framework for the collaborative inference that we investigate herein is the state-of-the-art by Wang et al. [9]. At a high level, this framework is comprised of two modules: one module on the local part

which performs the differential privacy noise-based perturbation in the inference phase; and the other module on the remote part, which conducts a noisy training process to mitigate the impact of noise perturbation on the inference accuracy performance of a DNN model.

As shown in Algorithm 1, the differential privacy based noise perturbation proceeds as follows. Given an input data sample x, the client passes it to the local part model and obtains $\tilde{\mathbf{x}} \leftarrow f_{\theta_1}(\mathbf{x})$. Then, noises sampled from the Laplace distribution are added to a bounded version of $\tilde{\mathbf{x}}$, producing the noisy intermediate output $\mathbf{x}^*$, which is sent to the cloud for inference. Note that bounding each value in $\tilde{\mathbf{x}}$ is needed because it is hard to directly estimate the global sensitivity of $f_{\theta_1}(\mathbf{x})$ for adding differential privacy noises. The bound B, as used in Algorithm 1, can be set as the median of the infinity norm with regard to a set of training examples during the training phase. We note that the client could optionally perform nullification on the input data sample $\mathbf{x}$ by randomly setting a portion of elements in $\mathbf{x}$ to zeros, masking some parts of $\mathbf{x}$ that are deemed highly sensitive.

As the perturbation will obviously degrade the accuracy performance of the DNN model, the design [9] constructively takes advantage of noisy training to fine-tune the remote part model $f_{\theta_2}(\cdot)$. The main idea is to perform training on both plain representations and noisy counterparts for the remote part model, taking into account the training losses for both plain representations and noisy representations. Here, a clear representation means the intermediate output obtained by passing an input data sample to the original clean local part model. We refer interested readers [9] for details on the algorithm for noisy training. Let $f'_{\theta_2}(\cdot)$ denote the fine-tuned remote part model. In the inference phase, upon receiving the noisy intermediate output from the client, the cloud conducts the inference by passing it to $f'_{\theta_2}(\cdot)$ and returns $f'_{\theta_2}(\mathbf{x}^*)$ to the client as the inference result.

Remark It is noted that since our goal is to evaluate the practical usability of the above state-of-the-art framework in the presence of the reconstruction attack, we exactly follow the Laplace mechanism-based construction in [9] and do not aim to propose new differential privacy mechanisms that can work for collaborative inference. We are aware that there are other mechanisms like the Gaussian mechanism and the exponential mechanism [13]. However, we emphasize that whether and how they can be effectively applied to the collaborative inference paradigm remains unclear. Indeed, it is non-trivial to apply differential privacy to the collaborative inference paradigm because simply adding noises locally will lead to poor utility of the inference service. This also accounts for why the prior work [9] needs to develop an algorithm for fine-tuning the model training process at the cloud serve, so as to balance privacy and utility. If there emerge other custom and workable differential privacy mechanisms for collaborative inference later, it would be interesting and valuable as well to explore their effectiveness against the reconstruction attack. In that case, we believe our initial study in this area can serve as good pointers and references.

Algorithm 2 Reconstruction Attack

Input: local part model f_{θ_1}; Intermediate output $f_{\theta_1}(x_0)$ for input x_0; Maximum number of iterations T; Hyperparameters λ for total variation and s for step size.

Output: Reconstructed $\widehat{x}$ for x_0

$L(x) = ||f_{\theta_1}(x) - f_{\theta_1}(x_0)||_2^2 + \lambda \cdot TV(x)$
$t = 0$
$x^{(0)} = \mathsf{Init}()$
While $(t < T)$ **do**
    $x^{(t+1)} = x^{(t)} - s \cdot \frac{\partial L(x^{(t)})}{\partial x^{(t)}}$
    $t = t + 1$
end
return $\widehat{x} = x^{(T)}$

2.4 Reconstruction attack against collaborative inference

In a recent work [10], He *et al.* proposed reconstruction attacks that allow the cloud to reconstruct the input image given the intermediate output and the local part model in the collaborative inference framework. Our study focuses on the reconstruction attack in the white-box setting because it is much stronger than that in the black-box setting. Evaluating differential privacy in the most powerful white-box attack setting arguably can better reflect how useful differential privacy can be in practice. For this attack setting, the local part model is known to the cloud, given that the whole DNN model is trained by the cloud, which also performs model splitting and provides the local part to the client. It is noted that the attack is proposed against images, so our evaluations are performed over image datasets. For other data types, we are not aware of any works that propose corresponding reconstruction attacks in the context of collaborative inference. Meanwhile, we note that the evaluation in the prior work [9] designing the differential privacy framework for collaborative inference is also dominated by image datasets. Further, it is worth noting that one main motivation for the collaborative inference paradigm initially proposed in [7] is to allow the local client to send to the cloud server a much smaller intermediate output rather than the large-sized raw input, for which image data as the input will benefit the most from such paradigm. Indeed in the seminal work [7], the evaluation is also conducted over image datasets.

Algorithm 2 gives the details of the studied reconstruction attack that aims to reconstruct input images in collaborative inference. Let x_0 denote an example input image and $\widehat{x}$ denote the reconstructed image against x_0. The

main idea is to formulate the reconstruction attack as an optimization problem under two requirements. Firstly, feeding $\widehat{x}$ to the local part model f_{θ_1} produces an intermediate output $f_{\theta_1}(\widehat{x})$ that is similar to the observed $f_{\theta_1}(x_0)$. Here the similarity is measured by the Euclidean distance. Secondly, $\widehat{x}$ is a natural image which follows the same distribution as the input samples for the DNN model. For this requirement, the total variation measure is adopted to enforce that the reconstructed image $\widehat{x}$ is as piece-wise smooth as possible.

3 Comprehensive evaluations

3.1 Experimental setup

Datasets. We use four datasets in our comprehensive empirical evaluations, including SVHN [14], GTSRB [15], CIFAR-10 [16], and STL-10 [17]. Figure 2 show the one class of each dataset. The overall specifications of these datasets are given in Table 1. It is noted that for each dataset, the clipping bound as shown in Table 1 is derived by computing the median of the infinity norms of intermediate outputs with regard to 100 randomly chosen training examples. Each dataset is introduced in more details below:

- SVHN. This dataset contains labeled images of house numbers in Google Street View images. Each image has a size of (32, 32, 3), and is labeled from 0 to 9. We randomly select 73, 200 images for training and 26, 000 for testing.
- GTSRB. This dataset contains labeled images of traffic sign images. The images have 3 channels but with varying sizes, and are categorized into more than 40 classes. There are more than 50, 000 images in total. In our evaluation, we randomly select 14, 600 images out of 10 classes for training and 4, 800 images for testing, with each image being resized to (32, 32, 3).

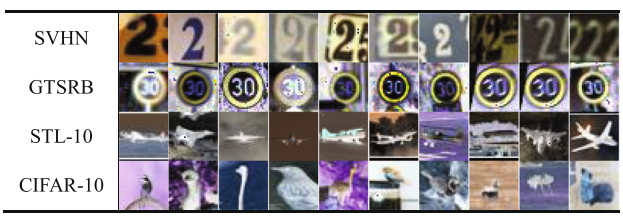


Fig. 2 Intra-class variation of each dataset: SVHN (digit number 2), GTSRB (30 km/h speed limit signs), CIFAR10 (bird), and STL10 (airplane). It is observable that SVHN < GTSRB < STL-10 < CIFAR-10 in terms of intra-class variation degree

- STL-10. This dataset contains labeled images of natural objects in 10 classes. There are 1, 300 images in each class. Each image has a size of (96, 96, 3). We randomly select 10, 000 images for training and 3, 000 images for testing, with each image being resized to (32, 32, 3).
- CIFAR-10. This dataset also contains labeled images of natural objects in 10 classes (such as airplane, bird, car, and cat), with 6, 000 images per class. Each image has a size of (32, 32, 3). There are 50, 000 training images and 10, 000 testing images, which are used in our evaluation.

3.1.1 Neural network architectures

The overall DNN architecture used in our evaluation is detailed in Fig. 3. Case 3 is the same as in [9][1] (where the local model contains 3 convolutional layers). We have considered more splitting cases: In Case 1, the local part model contains one convolutional layer; In Case 2, the number of local convolutional layers is 2. More details are given in Fig. 3.

The input size is (32, 32, 3), and the number of output class is 10. Following the prior work [9], we first derive the model parameters of the local part model (in different cases) from a pre-trained model over CIFAR-100 dataset, and then keep the local part model frozen for the client. That is, the local part model serves as a *generic feature extractor and is applicable to all different datasets* [9]. We trained the remote part model in a fine-tuned manner per each dataset which is introduced above (SVHN, GTSRB, STL-10, and CIFAR-10). Note that the input for the remote part model is the output obtained by feeding the data sample to the local part model.

3.1.2 Hyperparameters

For each dataset, we use the ADAM optimizer for training of the remote part model, following [9]. In order to determine the hyper-parameters, we follow the scale of the hyper-parameters in the prior work [9] as starting points, and then further fine-tune the hyper-parameters during our training process. The learning rate is set to 0.00001 for SVHN, 0.000002 for GTSRB, 0.0000027 for STL-10, and 0.00001 for CIFAR-10, respectively. The batch size being used is 300 for SVHN, 200 for GTSRB, 200 for STL-10, and 100 for CIFAR-10, respectively. The number of training epochs is 40 for SVHN, 100 for GTSRB, 500 for STL-10, and 100 for CIFAR-10, respectively. Similar to prior work related with the evaluation of differential

[1] Batch normalization is applied in our case to further improve the plain model accuracy.

Table 1 Specifications of datasets and clipping bounds under different splitting cases

Dataset	SVHN	GTSRB	STL-10	CIFAR-10
Training Set Size	73,200	14,600	10,000	50,000
Testing Set Size	26,000	4,800	3,000	10,000
Case 1 Bound	250.104	237.374	230.409	230.174
Case 2 Bound	7477.173	16385.918	12801.195	13058.996
Case 3 Bound	7774.149	9613.522	8346.818	10680.272

privacy in other contexts [18], we vary the privacy budget ϵ between 0.1 and 5000 which represents a wide range, and evaluate the results on accuracy and privacy strengths in the presence of the reconstruction attack. It is noted that the presented accuracy results are averaged over 5 runs.

3.1.3 Quantitative metrics

In addition to the visualization of reconstructed images, MSE, SSIM, and PSNR metrics are also adopted to quantify the reconstruction efficacy, which generally measures the difference between the original image and the reconstructed image.

Let A and B denote the original image and reconstructed image respectively, with the size of $m \times n$.

The pixel value at position (i, j) is denoted by $A(i, j)$ and $B(i, j)$ respectively for images A and B. In what follows we introduce each metric:

1. *Mean Squared Error* (MSE) measures the similarity between two images by computing the cumulative squared error of pixel values. The lower the value of MSE, the higher the similarity between two images. Specifically, it is computed via:

$$MSE(A,B) = \frac{1}{m \cdot n} \sum_{i,j=1,1}^{m,n} ||A(i,j) - B(i,j)||^2.$$

2. *Structural similarity* (SSIM) [19] is a perception-based metric which measures the similarity between two images based on structural information. It is computed as:

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_{AB} + C_2)}{(\mu_A^2 + \mu_B^2 + C_1)(\sigma_A^2 + \sigma_B^2 + C_2)},$$

where μ_A and μ_B are the mean value of pixels in image A and B, σ_A^2 and σ_B^2 are the variances, and σ_{AB} is the covariance, respectively. In addition, C_1 and C_2 are constants. The value of SSIM lies between the range of [0, 1], and a larger SSIM value indicates a higher similarity between two images.

3. *Peak signal-to-noise ratio* (PSNR) measures the similarity of two images via the peak error. Larger PSNR values indicate higher image similarity. It is computed via:

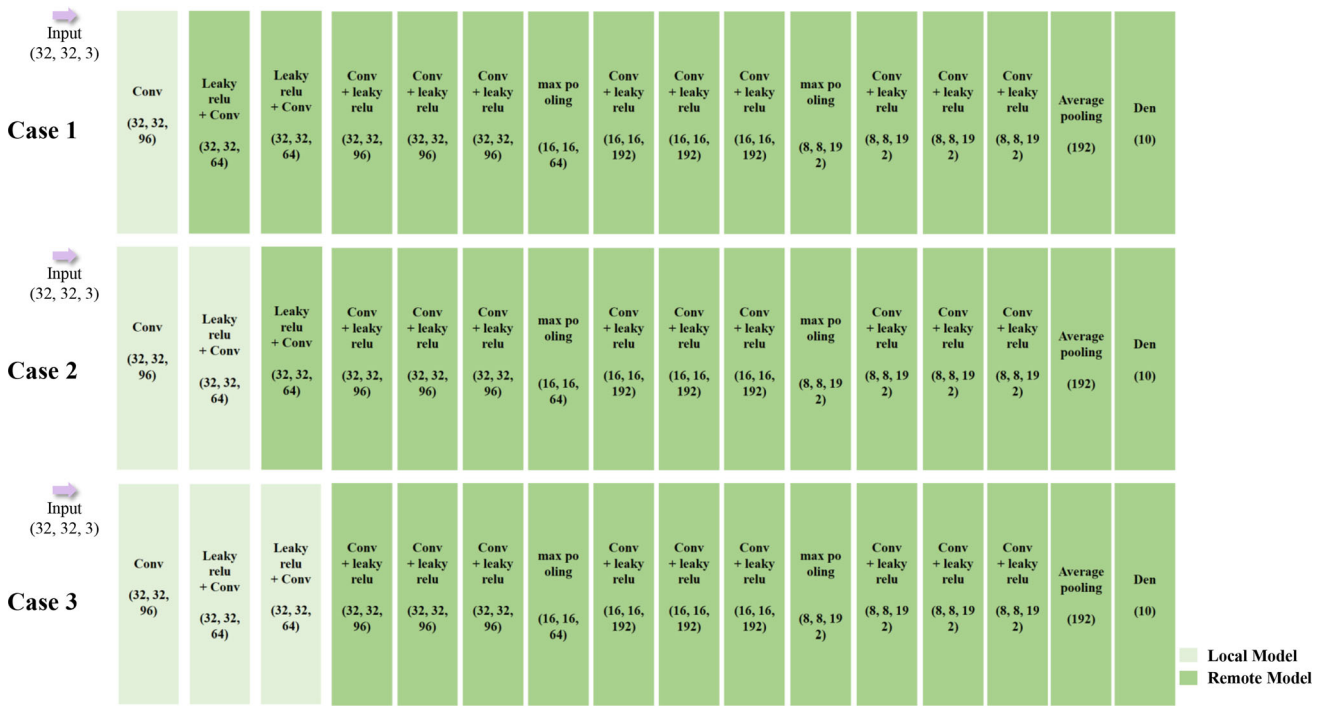


Fig. 3 The DNN architecture in our evaluation with varying splitting cases

$$PSNR(A,B) = 10\log_{10}(\frac{255^2}{MSE(A,B)}).$$

3.2 Results over the SVHN dataset

3.2.1 Utility under DP

As shown in Table 2, the baseline model over the SVHN dataset *without* differential privacy (DP) achieves accuracy of 93.498%, 92.695%, 92.953% in Cases 1, 2, and 3, respectively. In Fig. 4, we show the accuracy results of the DP method (Fig. 4a) as well as the normalized accuracy loss (Fig. 4b) against the baseline accuracy, under varying values of the privacy budget ϵ. As depicted in the figure, the DNN model under the DP method has essentially no utility for $\epsilon < 5$. For $\epsilon \geq 5$, the accuracy achieved by the DP method is rapidly getting close to the baseline accuracy. For instance, the accuracy results are 93.081%, 85.787%, 88.686% for $\epsilon = 5$ (normalized accuracy losses of 0.446%, 7.453%, 4.590%), 93.479%, 89.473%, 90.397% for $\epsilon = 10$ (normalized accuracy losses of 0.020%, 3.476%, 2.750%), 93.1928%, 91.8796%, 92.083% for $\epsilon = 100$ (normalized accuracy losses of 0.326%, 0.880%, 0.936%), and 93.199%, 92.083%, 92.249% for $\epsilon = 1000$ (normalized accuracy losses of 0.320%, 0.660%, 0.757%) for Case 1, Case 2, Case 3, respectively. These results show that on the SVHN dataset the DP method can still achieve good accuracy highly close to the baseline accuracy under suitable ϵ values.

3.2.2 Protection efficacy

We then examine the capability of the DP method in defending against the reconstruction attack. In Fig. 5, we show from a visual perspective the protection levels of differential privacy against the data reconstruction attack in Case 3 for some example testing images of the SVHN dataset. The results for Case 1 and 2 are shown in "Appendix A". That is, we show the original images and the reconstructed images derived by applying the attack to intermediate outputs of the local model part, with regard to varying privacy budget ϵ. As expected, the protection becomes less effective as the ϵ value increases. According to the visual results in the figure, no meaningful information can be observed from the reconstructed images for $\epsilon \leq 200$, indicating the DP method well protects the inputs against the reconstruction attack. For $\epsilon \geq 500$, the visual information of some images can be (clearly) observed from the reconstructed images, such as Sample 3 and Sample 4.

In Fig. 6, we show the evaluation of the results of the quantitative metrics (averaged over 100 randomly chosen testing images), including the MSE, SSIM, and PSNR, with regard to varying privacy budget ϵ. For the MSE metric, a clear descending trend is observed for $\epsilon < 10$. Then, the MSE values become relatively stable for $10 \leq \epsilon \leq 200$. For $\epsilon > 200$, the MSE values decreasingly evolve, indicating the reconstructed images due to the attack are getting closer to the original images. For the SSIM metric, overall there is an ascending trend, and a sharp increase can be observed for $\epsilon \geq 500$. Regarding the PSNR metric, we observe that the PSNR values remain almost stable regardless of the varying privacy budget ϵ. No clear ascending trends can be observed with the increase of the privacy budget ϵ (except when ϵ is greater than 1000). *This suggests that PSNR is not an appropriate metric for measuring the resistance of the DP method against the attack in this context.*

3.2.3 Note

From the above accuracy results and privacy measurement results, it is shown that on the SVHN dataset, the DNN model with the DP method, under suitable choices of ϵ values (e.g., $5 \leq \epsilon \leq 200$), can achieve accuracy comparable to the baseline while providing resistance to the reconstruction attack.

3.3 Results over the GTSRB dataset

3.3.1 Utility under DP

The baseline model over the GTSRB dataset *without* differential privacy (DP) achieves accuracy of 92.676%, 95.284%, 92.869% in Case 1, 2, and 3. Fig. 7 shows the accuracy results of the DP method (Fig. 7a) as well as the normalized accuracy loss (Fig. 7b) with respect to the baseline accuracy, under varying privacy budget ϵ. As depicted in the figure, the DNN model under the DP method has essentially no utility until ϵ exceeds 10 For $\epsilon \geq 10$, the accuracy achieved by the DP method is becoming close to the baseline accuracy. For instance, the accuracy results are 90.067%, 85.811%, 66.816% for $\epsilon = 10$ (normalized accuracy losses of 2.815%, 9.941%, 28.053%), 91.287%, 92.926%, 88.025% for $\epsilon = 100$ (normalized accuracy losses of 1.499%, 2.475%, 5.216%), and 91.533%, 92.535%, 89.587% for $\epsilon = 1000$ (normalized accuracy losses of 1.233%, 2.885%, 3.533%) in Case 1, Case 2, and Case 3, respectively. These results show that on the GTSRB dataset the accuracy loss due to the DP method is small under suitable ϵ values.

Table 2 Baseline accuracy without differential privacy

Model	SVHN	GTSRB	STL-10	CIFAR-10
Case 1	93.498%	92.676%	69.576%	82.932%
Case 2	92.695%	95.284%	67.448%	77.795%
Case 3	92.953%	92.869%	67.333%	84.500%
Average	93.049%	93.610%	68.119%	81.742%

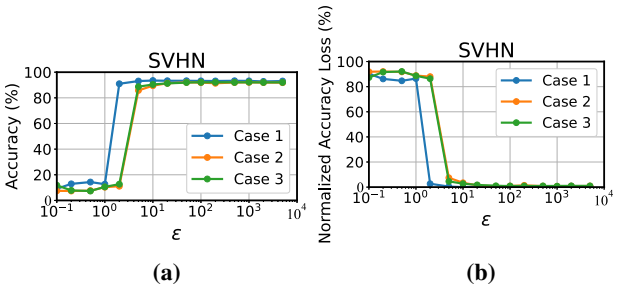


Fig. 4 Impact of the privacy budget ϵ on accuracy (SVHN)

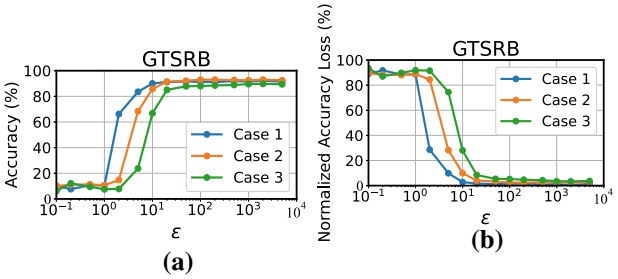


Fig. 7 Impact of the privacy budget ϵ on accuracy (GTSRB)

3.3.2 Protection efficacy

Figure 8 shows from a visual perspective the protection levels of the DP method against the data reconstruction attack in Case 3 for some example testing images of the GTSRB dataset. Case 1 and 2 are shown in "Appendix A". As expected, the protection becomes less effective with the increase of the ϵ value. According to the visual results in the figure, no meaningful information can be observed

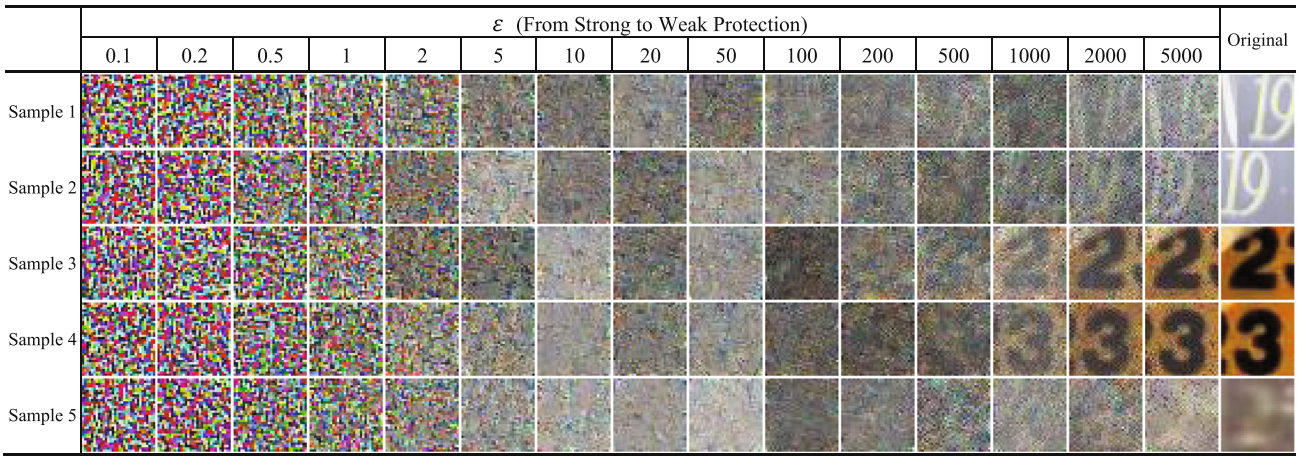


Fig. 5 Visual results of applying the attack against the DP method (SVHN Case 3)

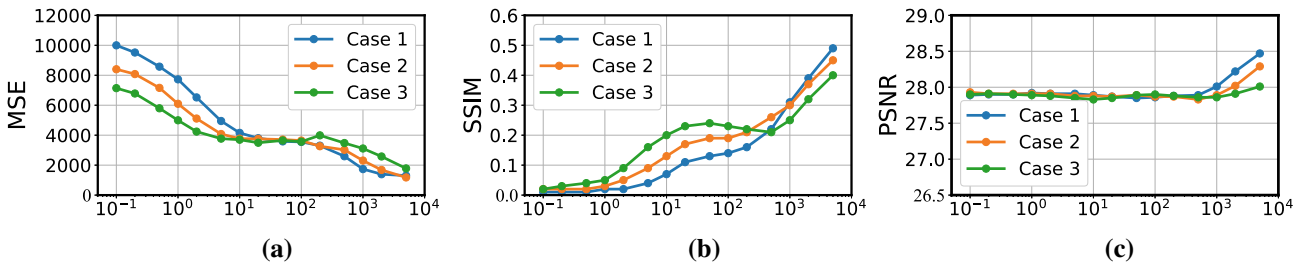


Fig. 6 Evaluation of the quantitative metrics for the reconstruction attack efficacy (SVHN): **a** MSE; **b** SSIM; **c** PSNR

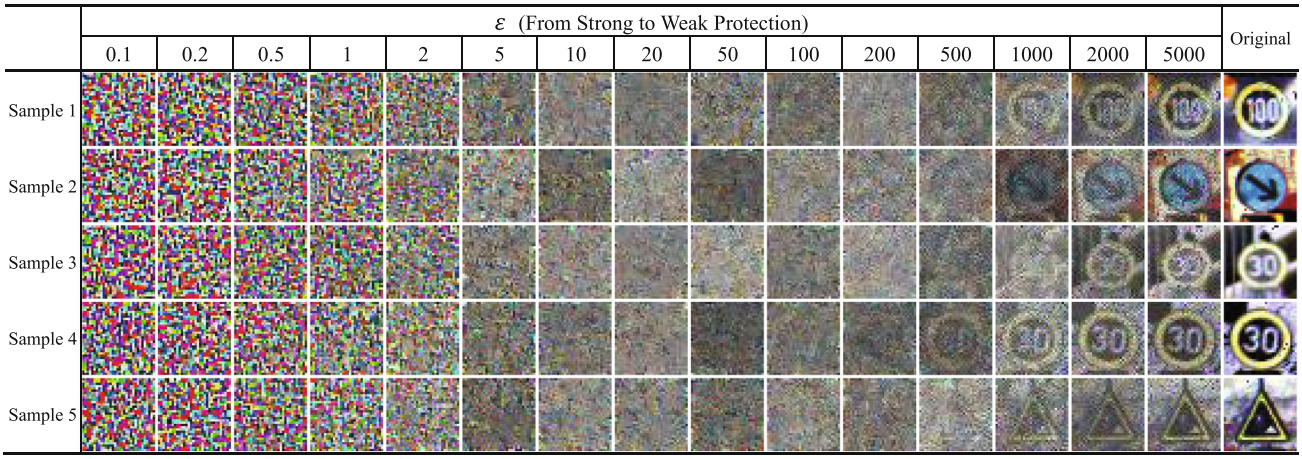


Fig. 8 Visual results of applying the attack against the DP method (GTSRB Case 3)

from the reconstructed images for $\epsilon \leq 200$, indicating the DP method well protects the inputs against the reconstruction attack. For $\epsilon \geq 500$, the visual information of the sample images can be (clearly) observed from the reconstructed images.

In Fig. 9, we show the evaluation of the results of the quantitative metrics (averaged over 100 randomly chosen testing images), including the MSE, SSIM, and PSNR, with regard to varying privacy budget ϵ. For the MSE metric, it reveals a clear descending trend for $\epsilon < 10$. Then, the MSE values become relatively stable for $10 \leq \epsilon \leq 200$. For $\epsilon > 200$, there is an obvious decrease in the MSE values, indicating the reconstructed images due to the attack are getting closer to the original images. For the SSIM metric, there is an overall ascending trend, and a dramatic increase is shown for $\epsilon \geq 500$. For the PSNR metric, we observe again that the PSNR values remain almost stable regardless of the privacy budget ϵ.

3.3.3 Note

From the above accuracy results and privacy measurement results, it is shown that over the GTSRB dataset, the DNN model with the DP method, under suitable choices of ϵ values (e.g., $100 \leq \epsilon \leq 200$), can achieve accuracy comparable to the baseline while providing resistance to the reconstruction attack.

3.4 Results over the STL-10 Dataset

3.4.1 Utility under DP

The baseline model over the STL-10 dataset *without* differential privacy (DP) achieves accuracy of 69.576%, 67.448%, 67.333% in Case 1, 2, and 3. Such accuracy levels also appeared in prior work [20], and is orthogonal to our study in this paper. In Fig. 10, we show the accuracy results of the DP method (Fig. 10a) as well as the normalized accuracy loss (Fig. 10b) against the baseline accuracy, under varying values of the privacy budget ϵ. As shown, the DNN model under the DP method has essentially no utility for $\epsilon < 10$. For $\epsilon \geq 10$, the accuracy achieved by the DP method is getting close to the baseline accuracy. For instance, the accuracy results are 64.690%, 50.149%, 57.593% for $\epsilon = 10$ (normalized accuracy losses of 7.023%, 25.648%, 14.465%), 66.207%, 63.003%, 62.598% for $\epsilon = 100$ (normalized accuracy losses of 4.842%, 6.590%, 7.032%), and 65.857%, 64.289%, 62.621% for $\epsilon = 1000$ (normalized accuracy losses of

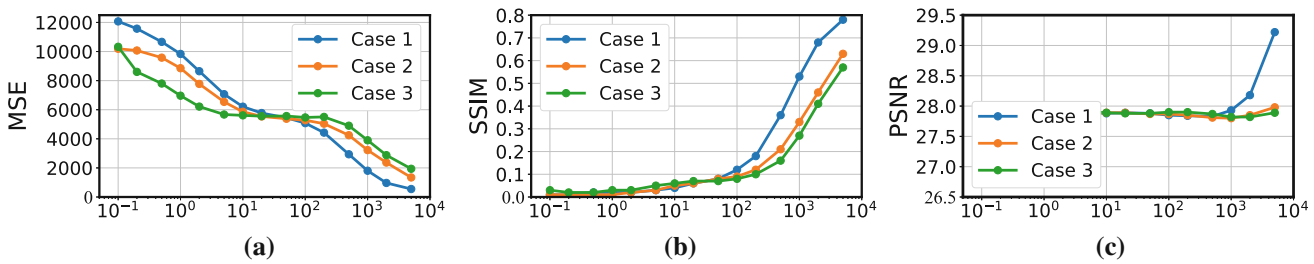


Fig. 9 Evaluation of the quantitative metrics for the reconstruction attack efficacy (GTSRB): **a** MSE; **b** SSIM; **c** PSNR

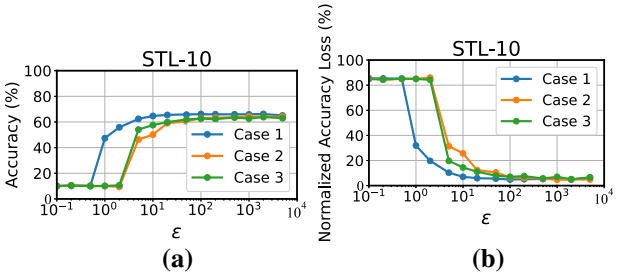


Fig. 10 Impact of the privacy budget ϵ on accuracy (STL-10)

5.345%, 4.684%, 6.998%) in Case 1, Case 2, and Case 3, respectively. These results show that on the STL-10 dataset, the DP method can achieve accuracy comparable to the baseline under suitable ϵ values.

3.4.2 Protection efficacy

Figure 11 shows from a visual perspective the protection levels of the DP method against the data reconstruction attack in Case 3 for some example testing images of the STL-10 dataset. Case 1 and 2 are shown in "Appendix A". As expected, the protection becomes less effective with the increase of the ϵ value. It is observed that even at $\epsilon = 1000$, the reconstructed images almost reveal no meaningful visual information of the original images. In Fig. 12, we show the evaluation of the results of the quantitative metrics (averaged over 100 randomly chosen testing images), including the MSE, SSIM, and PSNR, with regard to varying privacy budget ϵ. For the MSE metric, a clear descending trend is observed for $\epsilon < 10$. Then, the MSE values become relatively stable for $10 \leq \epsilon \leq 200$. For $\epsilon > 200$, the MSE values decreasingly evolve, indicating the reconstructed images due to the attack are getting closer to the original images. For the SSIM metric, overall there is an ascending trend, and a sharp increase can be observed for $\epsilon \geq 500$. Regarding the PSNR metric, it is shown again that the PSNR values remain almost stable regardless of the varying privacy budget ϵ.

3.4.3 Note

From the above accuracy results and privacy measurement results, it is shown that over the STL-10 dataset, the DNN model with the DP method, under suitable choices of ϵ values (e.g., $100 \leq \epsilon \leq 500$), can achieve accuracy comparable to the baseline while protecting the input privacy.

3.5 Results over the CIFAR-10 Dataset

3.5.1 Utility under DP

The baseline model over the CIFAR-10 dataset *without* differential privacy (DP) achieves accuracy of 82.932%, 77.795%, 84.5% in Case 1, 2, and 3. We show in Fig. 13 the accuracy results of the DP method (Fig. 13a) as well as the normalized accuracy loss (Fig. 13b) against the baseline accuracy, with regard to varying privacy budget ϵ. As shown in the figure, the DNN model under the DP method has almost no utility for $\epsilon < 50$. For $\epsilon \geq 200$, the accuracy does not increase significantly. In particular, for $200 \leq \epsilon \leq 1000$, the accuracy varies from 75.905%, 66.475%, 69.755% (normalized accuracy losses of 8.473%, 14.551%, 17.450%) to 76.588%, 65.462%, 72.114% (normalized accuracy losses of 7.650%, 15.853%, 14.658%), which is not close to the baseline accuracy of 82.932%, 77.795%, 84.5% in Case 1, Case 2, and Case 3, respectively. These results show that on the CIFAR-10 dataset, the DP method can retain meaningful utility of the DNN

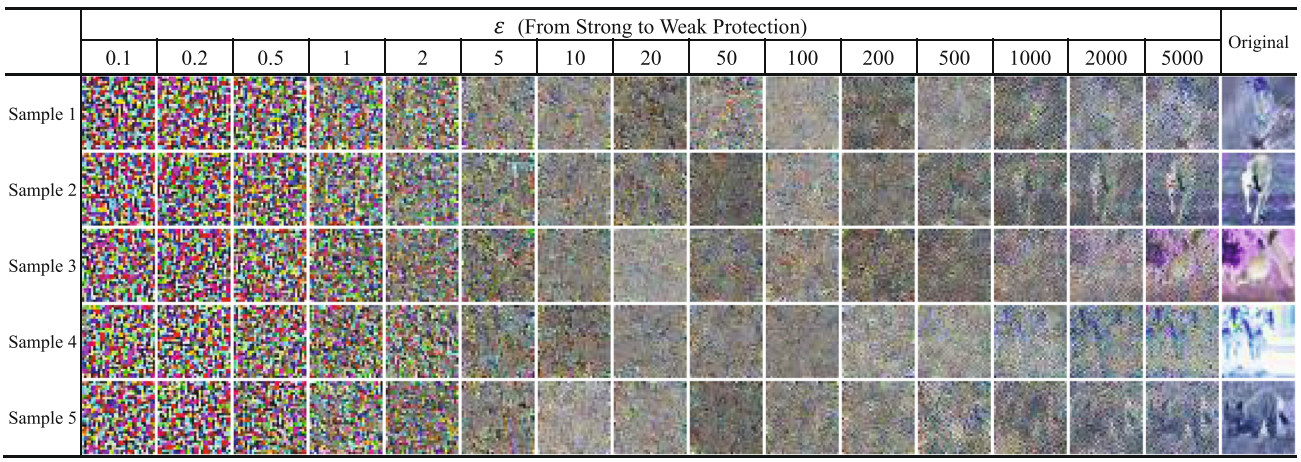


Fig. 11 Visual results of applying the attack against the DP method (STL-10 Case 3)

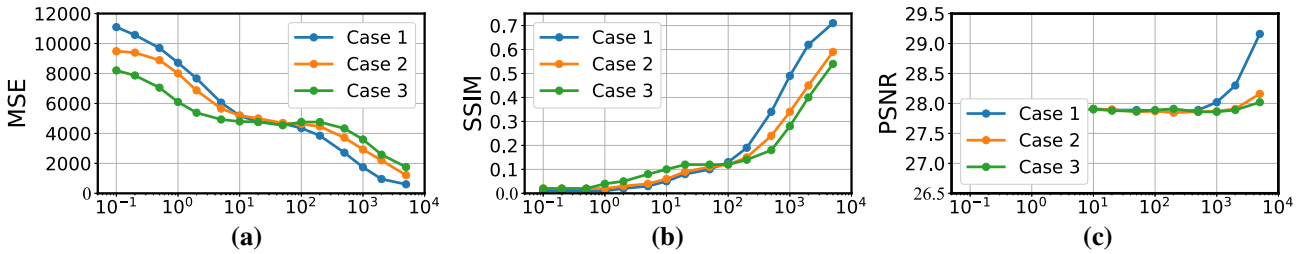


Fig. 12 Evaluation of the quantitative metrics for the reconstruction attack efficacy (STL-10): **a** MSE; **b** SSIM; **c** PSNR

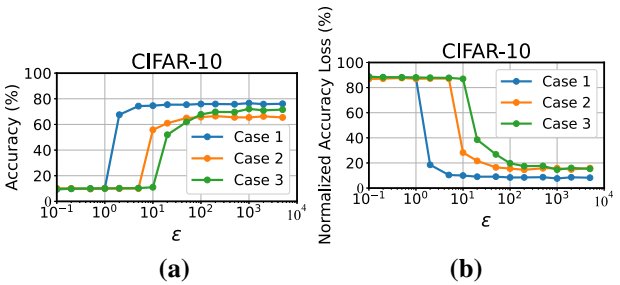


Fig. 13 Impact of the privacy budget ϵ on accuracy (CIFAR-10)

model yet the accuracy loss against the base accuracy is notable.

3.5.2 Protection efficacy

Figure 14 shows from a visual perspective the protection levels of the DP method against the data reconstruction attack in Case 3 for some example testing images of the CIFAR-10 dataset. Case 1 and 2 are shown in "Appendix A".

According to the visual results in the figure, no meaningful information can be observed from the reconstructed images for $\epsilon \leq 500$, indicating the DP method well protects the inputs against the reconstruction attack. Figure 15 shows the evaluation of the results of the quantitative metrics (averaged over 100 randomly chosen testing images), including the MSE, SSIM, and PSNR, with regard to varying privacy budget ϵ. For the MSE metric, a clear descending trend is observed for $\epsilon < 10$. Then, the MSE values become relatively stable for $10 \leq \epsilon \leq 200$. For $\epsilon > 200$, the MSE values decreasingly evolve, indicating the reconstructed images due to the attack are getting closer to the original images. For the SSIM metric, overall there is an ascending trend, and a sharp increase can be observed for $\epsilon \geq 500$. Regarding the PSNR metric, we observe that the PSNR values remain almost stable regardless of the varying ϵ.

3.5.3 Note

From the above accuracy results and privacy measurement results, it is shown that over the CIFAR-10 dataset, the DNN model with the DP method, under suitable choices of ϵ values (e.g., $200 \leq \epsilon \leq 500$), can only retain a meaningful utility of the DNN model while providing resistance to the reconstruction attack.

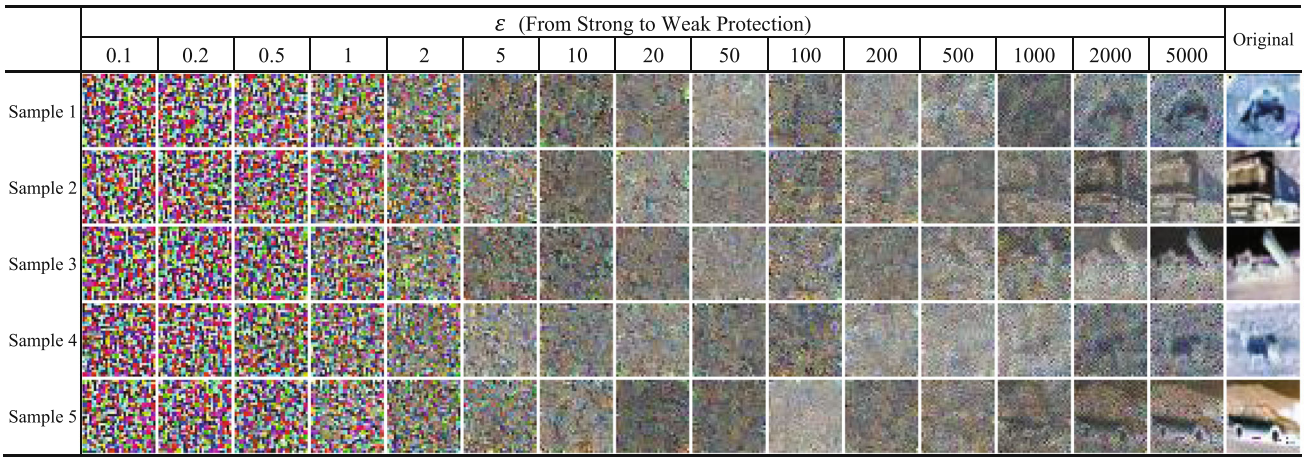


Fig. 14 Visual results of applying the attack against the DP method (CIFAR-10 Case 3)

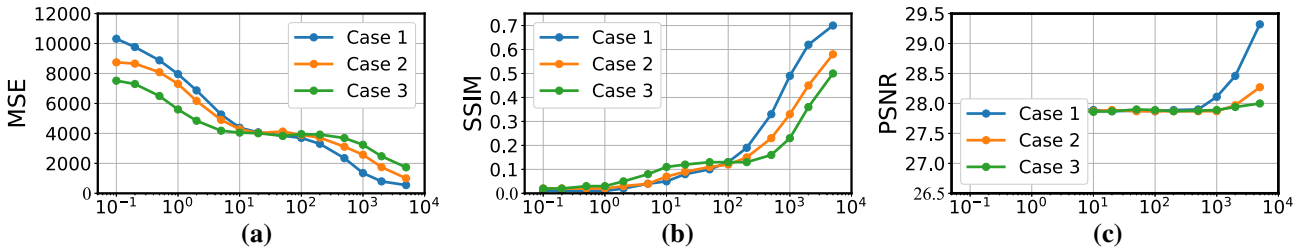


Fig. 15 Evaluation of the quantitative metrics for the reconstruction attack efficacy (CIFAR-10): **a** MSE; **b** SSIM; **c** PSNR

4 Insights and discussions

In response to our research question above on whether the differential privacy framework is able to protect collaborative inference while preserving utility, we discuss our findings and draw insights as follows.

Differential privacy is usable for collaborative inference in the presence of the data reconstruction attack From our results above, we consistently observe that the use of differential privacy can retain the (meaningful) usability of the DNN model, while providing protection on the input privacy in collaborative inference. For different datasets, however, our observation is that the suitable intervals of the privacy budget ϵ that can protect the input privacy while maintaining good accuracy could vary. For example, on the SVHN dataset, for $\epsilon = 5$, the (normalized) accuracy loss is 0.446%, 7.453%, 4.590% in Case 1, 2, and 3 while it is 9.854%, 28.257%, 74.435% in Case 1, 2, and 3 on the GTSRB dataset. On the GTSRB dataset, for $\epsilon = 500$, the visual information of original images can be observed from the reconstructed images, while no meaningful visual information from the attack can be observed on the CIFAR-10 dataset. Overall, across all the datasets being evaluated, our empirical observation is that the interval $100 \leq \epsilon \leq 200$ tends to provide a good trade-off between utility and privacy protection.

Whether differential privacy can achieve accuracy close to the baseline is dataset-dependent From the results over the four datasets, we observe that on the SVHN, GTSRB, and STL-10 datasets, the use of differential privacy is able to achieve accuracy close to the non-private baseline. For example, as shown in Fig. 16, for $\epsilon = 200$ where privacy protection is ensured, the (normalized) accuracy loss is 0.395%, 1.430%, 0.928% on SVHN, 1.671%, 2.464%, 4.720% on GTSRB, and 5.106%, 6.161%, 7.567% on STL-10 respectively, while it is up to 8.473%, 14.551%, 17.450% on CIFAR-10, for Case 1, Case 2, and Case 3, respectively.

On CIFAR-10, even when ϵ further increases to 2000 or 5000 where input privacy is compromised as shown in Fig. 14, the accuracy loss still stays at a high level, i.e., 8.581%, 14.794%, 16.024% for $\epsilon = 2000$, and 8.241%, 15.868%, 15.315% for $\epsilon = 5000$. Hence, we point out that even differential privacy can retain the (meaningful) usability of the DNN model in collaborative inference, it may not always be able to maintain the accuracy comparable to the non-private baseline.

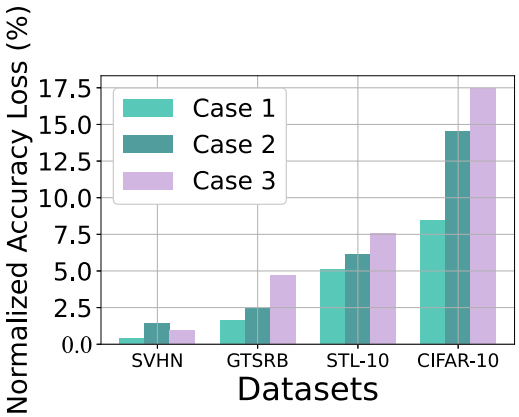


Fig. 16 Comparison of normalized accuracy drops over different datasets. (ϵ = 200)

Empirical guide Our empirical insight is that differential privacy appears to perform better for datasets with small intra-class variation in collaborative inference, since according to our observation CIFAR-10 has relatively large intra-class variation compared to the other datasets. Specifically, it is visually observable that the order of intra-class variation of the four tested datasets is as follows: CIFAR10>STL-10>GTSRB>SVHN. Accordingly, the averaged accuracy drops across different splitting cases due to differential privacy are 12.454%, 5.021%, 2.066%, 0.476% for CIFAR-10, STL-10, GTSRB, and SVHN, respectively, given the largest tested privacy budget per each dataset that can still provide protection against the reconstruction attack ($\epsilon = 200$ for SVHN and GTSRB, and $\epsilon = 500$ for STL-10 and CIFAR-10, as visually observed).

One simple criterion for intra-class variation is that the more specific the class is, the smaller the intra-class variation will be. For instance, the intra-class variation of German Shepherd Dog class is smaller than the intra-class of dog class, since the latter is more general. We hope our

initial study can stimulate research activities for further in-depth investigation.

Potential reason When the intra-class variation becomes larger, the sensitivity to the noise injected from the differential privacy could be higher. This could lead to notable degradation in the accuracy. A formal proof and corroboration in this direction is an interesting future work.

5 Related work

The user privacy issues have been extensively studied [21–29]. FakeMask [26] proposed a technology to protect users' privacy by disclosing fake contexts to solve the privacy problem on sensor-equipped smartphones. The work [21] proposed a privacy protection scheme based on a differential privacy model combined with clustering and randomization algorithms. In particular, there are privacy methods for machine learning models, such as [22–25, 27, 29]. A reinforcement learning algorithm that guarantees privacy in the optimization of the Markov decision-making process and can efficiently solve a large state space in a blockchain scenario by proposing a reinforcement learning-based offloading method was developed in [22]. The optimization method of the Deep Reinforcement Learning algorithm for detecting abnormal traffic that can monitor network transmission in real-time using anomaly detection and effectively detects external attacks is suggested in [24]. In addition, the works [25, 27, 30, 31] used federated learning for privacy protection in training models over distributed datasets. There is also a line of work [32, 33] on leveraging cryptographic techniques to secure DNN inference.

Our work is related to prior works on evaluating the effectiveness of differential privacy in machine learning with attacks. In [34], Rahman *et al.* evaluate membership inference attacks against a differentially private DNN model which is proposed in [35]. In [18], Jayaraman and Evans study the effectiveness of different relaxed notions of differential privacy which are proposed for training differentially private machine learning models, against membership inference attacks and attribute inference attacks. In [36], Bernau *et al.* compare local and central differential privacy mechanisms under membership inference attacks. All these works are proposed for the scenario where differential privacy is employed to protect the privacy of *training data*. Different from prior works, we present the first study on evaluating differential privacy when it is leveraged to protect the privacy of *model inputs* in collaborative inference, against the state-of-the-art data reconstruction attack.

6 Conclusion and future work

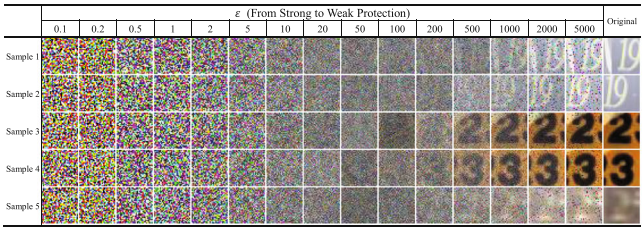
In this paper, we initiate the first comprehensive study on the assessment of the practical usability of differential privacy for collaborative inference in the presence of state-of-the-art data reconstruction attack. We conduct an extensive empirical evaluation over four datasets, examining the impact of varying privacy budget ϵ on the aspects including inference accuracy, visual protection strengths, and quantitative metrics. Our results reveal that differential privacy can be usable in the presence of the reconstruction attack under certain conditions. Practical insights and guidelines on the privacy-utility trade-offs have been drawn when deploying differential privacy for collaborative inference in practice. More specifically, an easy-to-adopt drawn guideline is that smaller intra-class variation of the dataset, more pragmatic of the DP for collaborative inference. We hope our work can lead to a deeper understanding of the effectiveness of using differential privacy for the protection of model input privacy in collaborative inference for IoT applications.

For furture work, it is interesting to explore quantitative measures for capturing dataset characteristics (e.g., intra-class variation) so as to better study the relation between dataset characteristics and the protection strengths of differntial privacy. It is also interesting to extend our study to non-image data, if reconstruction attacks against non-image data emerge in future.

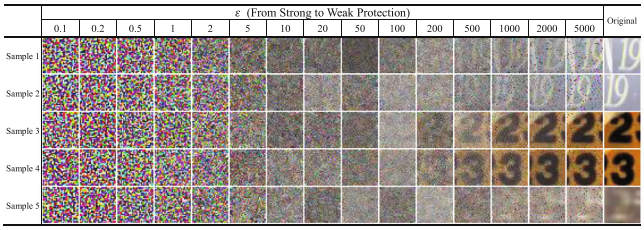
A More visual and quantitative evaluation results

Figure 17 show some visual evaluation results on Case 1 and Case 2 in datasets (SVHN, GTSRB, STL-10, CIFAR-10) regarding the protection levels of the DP method against the data reconstruction attack. We can see that the reconstruction attack is not effective even for smaller ϵ value as the local part model layer increases. It is observed that even at $\epsilon = 1000$ in Case 1, the reconstructed images reveal meaningful visual information of the original images, in Case 2, the reconstructed images, the reconstructed images almost reveal no meaningful information of the original images.

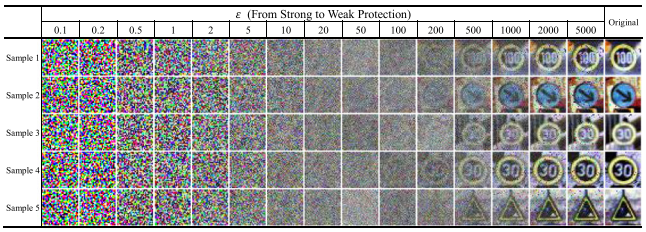
Tables 3, 4, 5, and 6 provide the quantitative evaluation results in terms of accuracy, MSE, SSIM, and PSNR. Note that the accuracy results were plotted in Figs. 4, 7, 10, and 13. And the MSE, SSIM, and PSNR results were plotted in Figs. 6, 9, 12, and 15. We provide the exact figures here to facilitate the observations.



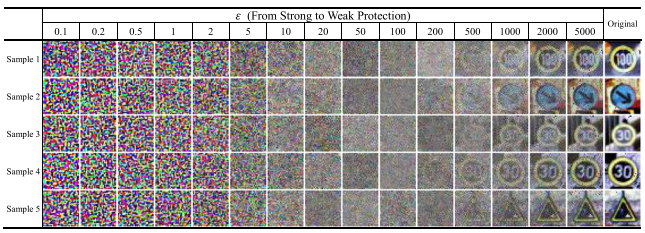
SVHN Case 1



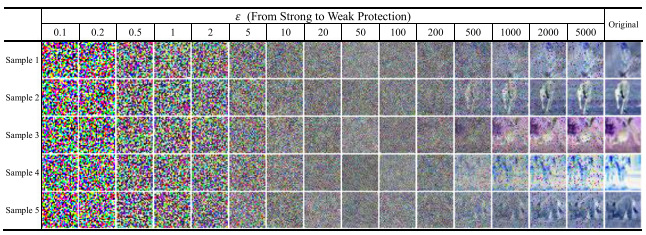
SVHN Case 2



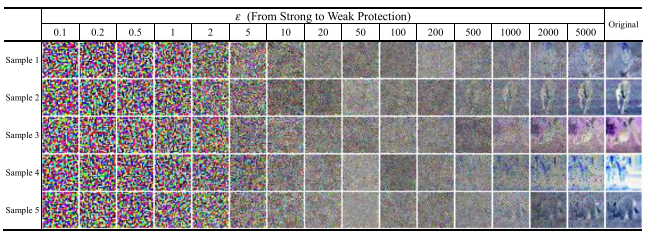
GTSRB Case 1



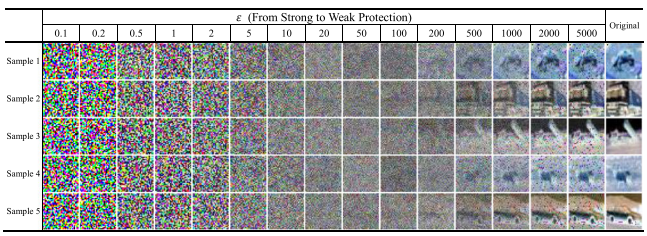
GTSRB Case 2



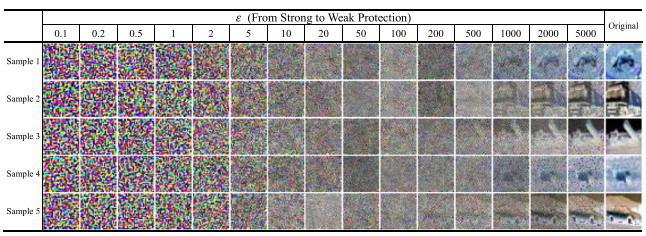
STL-10 Case 1



STL-10 Case 2



CIFAR-10 Case 1



CIFAR-10 Case 2

Fig. 17 Visual results of applying the attack against the DP method. (Case 1 and Case 2)

Table 3 Summary of quantitative evaluation results on SVHN

SVHN	ϵ	0.1	0.2	0.5	1	2	5	10	20
	Accuracy	9.086	12.916	14.319	12.673	90.977	93.081	93.479	93.311
Case 1	MSE	9997.76	9518.74	8574.61	7726.64	6521.06	4947.47	4153.4	3796.94
	SSIM	0.01	0.01	0.01	0.02	0.02	0.04	0.07	0.11
	PSNR	27.89	27.9	27.9	27.92	27.91	27.91	27.89	27.87
	Accuracy	7.492	7.366	7.635	10.336	11.108	85.787	89.473	91.364
Case 2	MSE	8401.52	8079.89	7158.49	6095.42	5111.37	4077.23	3766.98	3755.65
	SSIM	0.02	0.02	0.02	0.03	0.05	0.09	0.13	0.17
	PSNR	27.93	27.91	27.91	27.91	27.91	27.88	27.88	27.87
	Accuracy	11.820	7.795	7.338	10.563	12.673	88.686	90.397	91.290
Case 3	MSE	7140.53	6782.47	5795.56	4994.62	4239.78	3769.36	3696.93	3495.58
	SSIM	0.02	0.03	0.04	0.05	0.09	0.16	0.2	0.23
	PSNR	27.9	27.91	27.9	27.89	27.88	27.85	27.83	27.85

SVHN	ϵ	50	100	200	500	1000	2000	5000
	Accuracy	93.289	93.193	93.129	93.284	93.199	92.791	93.072
Case 1	MSE	3585.67	3550.77	3301.97	2593.92	1737.37	1394.46	1302.65
	SSIM	0.13	0.14	0.16	0.22	0.31	0.39	0.49
	PSNR	27.85	27.86	27.88	27.89	28.01	28.22	28.47
	Accuracy	92.090	91.880	91.370	91.963	92.083	91.925	91.853
Case 2	MSE	3707.14	3640.8	3262.77	3023.88	2307.26	1679.77	1179.61
	SSIM	0.19	0.19	0.21	0.26	0.3	0.37	0.45
	PSNR	27.89	27.87	27.87	27.83	27.89	28.02	28.29
	Accuracy	91.974	92.083	92.097	92.027	92.249	92.017	92.031
Case 3	MSE	3649.12	3566.92	3989.44	3476.29	3112.32	2578.59	1781.42
	SSIM	0.24	0.23	0.22	0.21	0.25	0.32	0.4
	PSNR	27.89	27.9	27.88	27.86	27.86	27.91	28.01

Table 4 Summary of quantitative evaluation results on GTSRB

GTSRB	ϵ	0.1	0.2	0.5	1	2	5	10	20
	Accuracy	10.201	7.654	10.663	10.540	66.166	83.544	90.067	91.141
Case 1	MSE	12071.39	11571.36	10665.28	9834.47	8647.24	7068.13	6209.64	5771.9
	SSIM	0.01	0.01	0.02	0.02	0.02	0.03	0.04	0.06
	PSNR	27.91	27.89	27.9	27.9	27.89	27.89	27.88	27.89
	Accuracy	10.129	11.083	11.381	10.628	14.764	68.360	85.811	91.411
Case 2	MSE	10188.53	10071.2	9580.27	8853.6	7773.82	6539.38	5858.98	5538.74
	SSIM	0.01	0.01	0.01	0.01	0.02	0.03	0.05	0.06
	PSNR	27.91	27.88	27.9	27.9	27.91	27.88	27.89	27.89
	Accuracy	6.199	12.089	9.442	7.525	7.832	23.742	66.816	85.033
Case 3	MSE	10325.88	8601.28	7800.56	6964.68	6215.12	5670.55	5612.39	5556.87
	SSIM	0.03	0.02	0.02	0.03	0.03	0.05	0.06	0.07
	PSNR	27.87	27.91	27.9	27.9	27.9	27.91	27.89	27.88

GTSRB	ϵ	50	100	200	500	1000	2000	5000
	Accuracy	91.565	91.287	91.128	91.938	91.533	92.236	91.452
Case 1	MSE	5444.03	5078.06	4425.78	2936.96	1805.7	966.69	548.52
	SSIM	0.08	0.12	0.18	0.36	0.53	0.68	0.78
	PSNR	27.88	27.85	27.84	27.83	27.93	28.18	29.22
	Accuracy	92.072	92.926	92.936	92.711	92.535	93.034	92.522
Case 2	MSE	5379.79	5277.89	5039.41	4247.2	3224.56	2366.67	1343.37
	SSIM	0.08	0.09	0.12	0.21	0.33	0.46	0.63
	PSNR	27.87	27.87	27.85	27.81	27.8	27.85	27.98
	Accuracy	87.855	88.025	88.486	88.858	89.587	89.756	89.417
Case 3	MSE	5564.47	5468.64	5505.66	4901.11	3900.2	2875.03	1939.66
	SSIM	0.07	0.08	0.1	0.16	0.27	0.41	0.57
	PSNR	27.88	27.9	27.9	27.87	26.81	27.82	27.89

Table 5 Summary of quantitative evaluation results on STL-10

STL-10	ϵ	0.1	0.2	0.5	1	2	5	10	20
	Accuracy	10.105	10.183	10.317	47.362	55.837	62.393	64.670	65.461
Case 1	MSE	11095.13	10570.25	9701.28	8717.42	7668.91	6058.02	5181.42	4764.48
	SSIM	0.01	0.01	0.01	0.01	0.02	0.03	0.05	0.08
	PSNR	27.88	27.89	27.89	27.89	27.91	27.89	27.9	27.89
	Accuracy	9.692	10.602	10.115	10.099	9.443	46.263	50.149	59.110
Case 2	MSE	9487.52	9394.37	8887.52	8006.75	6872.39	5656.2	5194.35	4999.85
	SSIM	0.02	0.02	0.02	0.02	0.03	0.04	0.06	0.09
	PSNR	27.89	27.88	27.87	27.9	27.9	27.88	27.91	27.9
	Accuracy	10.177	10.460	9.958	10.105	10.622	54.072	57.593	59.896
Case 3	MSE	8202.77	7868.32	7055.42	6097.01	5379.97	4930.15	4795.49	4763.31
	SSIM	0.02	0.02	0.02	0.04	0.05	0.08	0.1	0.12
	PSNR	27.9	27.9	27.9	27.9	27.89	27.88	27.91	27.88

STL-10	ϵ	50	100	200	500	1000	2000	5000
	Accuracy	65.693	66.207	66.023	65.935	65.857	66.232	65.055
Case 1	MSE	4661.07	4352.32	3849.4	2714.01	1742.79	953.52	602.52
	SSIM	0.1	0.13	0.19	0.34	0.49	0.62	0.71
	PSNR	27.89	27.88	27.85	27.89	28.02	28.3	29.16
	Accuracy	60.373	63.003	63.286	63.707	64.289	64.098	64.178
Case 2	MSE	4688.91	4622.9	4468.96	3711.47	2927.61	2201.92	1212.14
	SSIM	0.11	0.12	0.15	0.24	0.34	0.45	0.59
	PSNR	27.86	27.87	27.85	27.86	27.87	27.91	28.16
	Accuracy	61.983	62.598	62.238	63.508	62.621	63.872	62.952
Case 3	MSE	4539.6	4760.15	4757.22	4331.99	3597.64	2583.19	1752.03
	SSIM	0.12	0.12	0.14	0.18	0.28	0.4	0.54
	PSNR	27.87	27.89	27.91	27.86	27.86	27.89	28.02

Table 6 Summary of quantitative evaluation results on CIFAR-10

CIFAR-10	ϵ	0.1	0.2	0.5	1	2	5	10	20
	Accuracy	9.376	9.865	9.953	10.256	67.578	74.290	74.645	75.424
Case 1	MSE	10316.43	9765.56	8875.63	7957.14	6869.52	5264.82	4380.35	4072.72
	SSIM	0.01	0.01	0.01	0.01	0.02	0.04	0.05	0.08
	PSNR	27.88	27.91	27.9	27.9	27.91	27.91	27.89	27.87
	Accuracy	10.245	10.027	9.605	10.097	9.958	9.985	55.798	60.954
Case 2	MSE	8742.84	8657.68	8084.87	7303.02	6171.18	4902.38	4277.36	4028.51
	SSIM	0.02	0.02	0.02	0.02	0.03	0.04	0.07	0.09
	PSNR	27.9	27.9	27.91	27.9	27.9	27.89	27.87	27.89
	Accuracy	9.621	9.882	9.985	10.058	10.221	10.339	11.006	51.949
Case 3	MSE	7518.53	7298.3	6497.63	5598.26	4845.48	4179.38	4051.71	4013.06
	SSIM	0.02	0.02	0.03	0.03	0.05	0.08	0.11	0.12
	PSNR	27.91	27.89	27.9	27.9	27.89	27.87	27.86	27.87

CIFAR-10	ϵ	50	100	200	500	1000	2000	5000
	Accuracy	75.395	75.940	75.905	75.756	76.588	75.816	76.098
Case 1	MSE	3843.67	3698.5	3302.37	2346.79	1353.97	795.22	550.79
	SSIM	0.1	0.13	0.19	0.33	0.49	0.62	0.7
	PSNR	27.88	27.87	27.89	27.9	28.11	28.46	29.32
	Accuracy	64.976	65.707	66.475	65.541	65.462	66.286	65.451
Case 2	MSE	4123.44	3897.66	3715.12	3113.49	2576.85	1750.36	1002
	SSIM	0.11	0.12	0.15	0.23	0.33	0.45	0.58
	PSNR	27.87	27.87	27.87	27.87	27.87	27.97	28.27
	Accuracy	61.840	67.829	69.755	69.610	72.114	70.960	71.559
Case 3	MSE	3836.35	3954.54	3921.73	3689.95	3241.79	2475.87	1734.69
	SSIM	0.13	0.13	0.13	0.16	0.23	0.36	0.5
	PSNR	27.9	27.89	27.87	27.88	27.89	27.94	28

Acknowledgements This paper was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515110027, in part by the Shenzhen Science and Technology Program under Grant RCBS20210609103056041, in part by the National Natural Science Foundation of China under Grant 62002167, in part by the Natural Science Foundation of JiangSu under Grant BK20200461, in part by the Research Grants Council of Hong Kong under Grants CityU 11217819, 11217620, RFS2122-1S04, N_CityU139/21, C2004-21GF, R1012-21, and R6021-20F, in part by the Shenzhen Municipality Science and Technology Innovation Commission under Grant SGDX20201103093004019, and in part by the Information & communications Technology Promotion grant funded by the Korea government.

Author Contributions Conceptualization: Jihyeon Ryu, Yifeng Zheng, Yansong Gao, Alsharif Abuadbba; Methodology: Jihyeon Ryu, Yifeng Zheng, Yansong Gao, Alsharif Abuadbba; Formal analysis and investigation: Jihyeon Ryu, Yifeng Zheng, Yansong Gao; Writing—original draft preparation: Jihyeon Ryu, Yifeng Zheng, Yansong Gao, Alsharif Abuadbba; Writing - review and editing: Junyaup Kim, Dongho Won, Surya Nepal, Hyoungshick Kim, Cong Wang; Funding acquisition: Yifeng Zheng, Yansong Gao.

Funding This paper was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515110027, in part by the Shenzhen Science and Technology Program under Grant RCBS20210609103056041, in part by the National Natural Science Foundation of China under Grant 62002167, in part by the Natural Science Foundation of JiangSu under Grant BK20200461, in part by the Research Grants Council of Hong Kong under Grants CityU 11217819, 11217620, RFS2122-1S04, N_CityU139/21, C2004-21GF, R1012-21, and R6021-20F, in part by the Shenzhen Municipality Science and Technology Innovation Commission under Grant SGDX20201103093004019, and in part by the Information & communications Technology Promotion grant funded by the Korea government.

Data Availibility Statement The datasets used during this study are publicly available, and the references to their sources have been given in this published article.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval This article does not contain any studies with human participants performed by any of the authors.

References

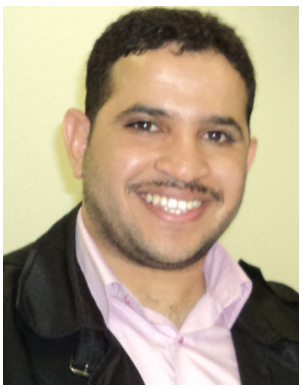
1. Yao, S., Hu, S., Zhao, Y., Zhang, A., & Abdelzaher, T. F. (2017). Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of WWW*.
2. Radu, V., Tong, C., Bhattacharya, S., Lane, N. D., Mascolo, C., Marina, M. K., & Kawsar, F. (2017). Multimodal deep learning for activity and context recognition. In *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, Vol. 1, no. 4, pp. 157:1–157:27.
3. Yao, S., Zhao, Y., Shao, H., Zhang, A., Zhang, C., Li, S., & Abdelzaher, T. F. (2017) "Rdeepsense: Reliable deep mobile computing models with uncertainty estimations," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, Vol. 1, no. 4, pp. 173:1–173:26.
4. Yao, S., Zhao, Y., Shao, H., Zhang, C., Zhang, A., Hu, S., Liu, D., Liu, S., Su, L., & Abdelzaher, T. F. (2018). Sensegan: Enabling deep learning for internet of things with a semi-supervised framework. In *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, Vol. 2, no. 3, pp. 144:1–144:21.
5. Yao, S., Zhao, Y., Zhang, A., Hu, S., Shao, H., Zhang, C., Su, L., & Abdelzaher, T. (2018). Deep learning for the internet of things. *Computer, 51*(5), 32–41.
6. Yao, S., Zhao, Y., Shao, H., Liu, S., Liu, D., Su, L., & Abdelzaher, T. F. (2018). Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices. In *Proceedings of ACM SenSys*.
7. Teerapittayanon, S., McDanel, B., & Kung, H. T. (2017). Distributed deep neural networks over the cloud, the edge and end devices. In *Proceedings of IEEE ICDCS*.
8. Ko, J. H., Na, T., Amir, M. F., & Mukhopadhyay, S. (2018). Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained internet-of-things platforms. In *Proceedings of IEEE international conference on advanced video and signal based surveillance*.
9. Wang, J., Zhang, J., Bao, W., Zhu, X., Cao, B., & Yu, P. S. (2018). Not just privacy: Improving performance of private deep learning in mobile cloud. In *Proceedings of KDD*.
10. He, Z., Zhang, T., & Lee, R. B. (2019). Model inversion attacks against collaborative inference. In *Proceedings of ACSAC*.
11. Dwork, C. (2006). Differential privacy. In *Proceedings of ICALP*.
12. Dwork, C., McSherry, F., Nissim, K., & Smith, A. D. (2006). Calibrating noise to sensitivity in private data analysis. In *Proceedings of TCC*.
13. Bai, J., Li, Y., Li, J., Yang, X., Jiang, Y., & Xia, S. (2022). Multinomial random forest. *Pattern Recognition, 122*, 108331.
14. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *ICLR AI for social good workshop*.
15. Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks, 32*, 323–332.
16. Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Tech. Rep.
17. Coates, A., Ng, A. Y., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of AISTATS*.
18. Jayaraman, B., & Evans, D. (2019). Evaluating differentially private machine learning in practice. In *Proceedings of USENIX security*.
19. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing, 13*(4), 600–612.
20. Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. In *Proceedings of NeurIPS*, pp. 766–774.
21. Huang, H., Zhang, D., Xiao, F., Wang, K., Gu, J., & Wang, R. (2020). Privacy-preserving approach pbcn in social network with differential privacy. *IEEE Transactions on Network and Service Management, 17*(2), 931–945.
22. Nguyen, D. C., Pathirana, P. N., Ding, M., & Seneviratne, A. (2020). Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning. *IEEE Transactions on Network and Service Management, 17*(4), 2536–2549.
23. Andreoletti, D., Velichkova, T., Verticale, G., Tornatore, M., & Giordano, S. (2020). A privacy-preserving reinforcement learning algorithm for multi-domain virtual network embedding. *IEEE Transactions on Network and Service Management, 17*(4), 2291–2304.
24. Dong, S., Xia, Y., & Peng, T. (2021). Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. *IEEE Transactions on Network and Service Management*.
25. Khan, L. U., Han, Z., Niyato, D., & Hong, C. S. (2021). Socially-aware-clustering-enabled federated learning for edge networks. *IEEE Transactions on Network and Service Management*.
26. Zhang, L., Cai, Z., & Wang, X. (2016). Fakemask: A novel privacy preserving approach for smartphones. *IEEE Transactions on Network and Service Management, 13*(2), 335–348.
27. Subramanya, T., & Riggio, R. (2021). Centralized and federated learning for predictive vnf autoscaling in multi-domain 5g networks and beyond. *IEEE Transactions on Network and Service Management, 18*(1), 63–78.
28. Ding, W., Hu, R., Yan, Z., Qian, X., Deng, R. H., Yang, L. T., & Dong, M. (2019). An extended framework of privacy-preserving computation with flexible access control. *IEEE Transactions on Network and Service Management, 17*(2), 918–930.
29. Groleat, T., & Pouyllau, H. (2012). Distributed learning algorithms for inter-nsp sla negotiation management. *IEEE Transactions on Network and Service Management, 9*(4), 433–445.
30. Zheng, Y., Lai, S., Liu, Y., Yuan, X., Yi, X., & Wang, C. (2022). Aggregation service for federated learning: An efficient, secure, and more resilient realization. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2022.3146448.
31. Zhu, L., Liu, X., Li, Y., Yang, X., Xia, S., & Lu, R. (2022)"A fine-grained differentially private federated learning against leakage from gradients," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11 500–11 512.
32. Zheng, Y., Duan, H., Tang, X., Wang, C., & Zhou, J. (2021). Denoising in the dark: Privacy-preserving deep neural network-based image denoising. *IEEE Transactions on Dependable and Secure Computing, 18*(3), 1261–1275.
33. Liu, X., Zheng, Y., Yuan, X., & Yi, X. (2021). Medisc: Towards secure and lightweight deep learning as a medical diagnostic service. In *Proceedings of ESORICS*.
34. Rahman, M. A., Rahman, T., Laganière, R., & Mohammed, N. (2018). Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy, 11*(1), 61–79.
35. Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of ACM CCS*.

36. Bernau, D., Grassal, P., Robl, J., & Kerschbaum, F. (2019). Assessing differentially private deep learning with membership inference. *CoRR*, Vol. abs/1912.11328.

Jihyeon Ryu received the B.S. degree in Mathematics and Computer Science from Sungkyunkwan University, South Korea, in 2018. She is currently pursuing the Ph.D. degree in the Department of Computer Science and Engineering at the Sungkyunkwan University. Her current research interests include cyber security, machine learning, and user authentication.

Yifeng Zheng is an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2019. He worked as a postdoc with the Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia and City University of Hong Kong. His work has appeared in prestigious conferences (such as ESORICS, DSN, and ACM AsiaCCS) and journals (such as IEEE TDSC, TKDE, TCAD, TIFS, and TSC). He received the Best Paper Award in the European Symposium on Research in Computer Security (ESORICS) 2021. His current research interests are focused on security and privacy related to cloud computing, IoT, machine learning, and multimedia.

Yansong Gao received his Ph.D. degree from the University of Adelaide, Australia, in 2017. He was with Data61, CSIRO, Sydney, Australia as a Postdoc Research Fellow. He is now with Nanjing University of Science and Technology, China, as an associate professor. His current research interests are AI security and privacy, hardware security, and system security.



Alsharif Abuadbba is Senior Research Scientist at CSIRO's Data61, Australia. Alsharif has joined Data61 Distributed System Security group early 2019 as a Research Scientist and Cybersecurity CRC fellow.

Junyaup Kim is a master student at Sungkyunkwan University. His research interests lie in media forensics, speech recognition, audio engineering and distributed systems. He received the Next Generation Leader award by Korea Professional Engineer Association (2019) and the highest prize by National Research Foundation of Korea by developing Wafer edge inspection system in X-corps festival (2019).

Dongho Won received B.S., M.S. and Ph.D. in Electronic Engineering from Sungkyunkwan University, South Korea. After working in Electronics and Telecommunication Research Institute for two years, he joined Sungkyunkwan University. He also served as a President of Korea Institute of Information Security and Cryptography. His research interests are cryptology and information security.

Surya Nepal is a Senior Principal Research Scientist at Data61. He currently leads the distributed systems security group. His main research focus is on the development and implementation of technologies in the area of distributed systems (including cloud, IoT and edge computing) and social networks, with a specific focus on security, privacy, and trust. He has more than 200 peer-reviewed publications to his credit. He has co-edited three books, including security, privacy, and trust in cloud systems by Springer, and co-invented three patents. He is a member of the editorial boards of IEEE Transactions on Service Computing, ACM Transactions on Internet Technology and Frontiers of Big Data-Security Privacy, and Trust. He is currently a theme leader of the cyber security Cooperative Research Centre (CRC), a national initiative in Australia. He holds conjoint faculty position at UNSW and an honorary professor position at Macquarie University.

Hyoungshick Kim received the B.S. degree from the Department of Information Engineering, Sungkyunkwan University, in 1999, the M.S. degree from the Department of Computer Science, KAIST, in 2001, and the Ph.D. degree from the Computer Laboratory, University of Cambridge, in 2012. He was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, The University of British Columbia. He was a Senior Engineer with Samsung Electronics from 2004 to 2008. He also served as a member of DLNA and Coral Standardization for DRM interoperability in home networks. He is currently an Associate Professor with the Department of Computer Science and Engineering, College of Software, Sungkyunkwan University. His current research interests are focused on social computing and usable security.

Cong Wang is a Professor in the Department of Computer Science, City University of Hong Kong. His research interests include data and network security, blockchain and decentralized applications, and privacy-enhancing technologies. He has been conferred the RGC Research Fellow in 2021. He received the Outstanding Researcher Award (junior faculty) in 2019, the Outstanding Supervisor Award in 2017 and the President's Awards in 2019 and 2016, all from City University of Hong Kong. He is a co-recipient of the Best Paper Award of IEEE ICDCS 2020, ICPADS 2018, MSN 2015, the Best Student Paper Award of IEEE ICDCS 2017, and the IEEE INFOCOM Test of Time Paper Award 2020. His research has been supported by multiple government research fund agencies, including National Natural Science Foundation of China, Hong Kong Research Grants Council, and Hong Kong Innovation and Technology Commission. He has served as associate editor for IEEE Transactions on Dependable and Secure Computing (TDSC), IEEE Transactions on Services Computing (TSC), IEEE Internet of Things Journal (IoT-J), IEEE Networking Letters, and The Journal of Blockchain Research, and TPC co-chairs for a number of IEEE conferences and workshops. He is a fellow of the IEEE, and member of the ACM.