Can Differential Privacy Practically Protect Collaborative Deep Learning Inference for the Internet of Things?

Jihyeon Ryu, Yifeng Zheng, Yansong Gao, Sharif Abuadbba, Junyaup Kim, Dongho Won, Surya Nepal, Hyoungshick Kim, and Cong Wang, Fellow, IEEE

Abstract—Collaborative inference has recently emerged as an intriguing framework for applying deep learning to Internet of Things (IoT) applications, which works by splitting a DNN model into two subpart models respectively on resource-constrained IoT devices and the cloud. Even though IoT applications’ raw input data is not directly exposed to the cloud in such framework, revealing the local-part model’s intermediate output still entails privacy risks. For mitigation of privacy risks, differential privacy could be adopted in principle. However, the practicality of differential privacy for collaborative inference under various conditions remains unclear. For example, it is unclear how the calibration of the privacy budget $\epsilon$ will affect the protection strength and model accuracy in presence of the state-of-the-art reconstruction attack targeting collaborative inference, and whether a good privacy-utility balance exists. In this paper, we provide the first systematic study to assess the effectiveness of differential privacy for protecting collaborative inference in presence of the reconstruction attack, through extensive empirical evaluations on various datasets. Our results show differential privacy can be used for collaborative inference when confronted with the reconstruction attack, with insights provided about privacy-utility trade-offs. Specifically, across the evaluated datasets, we observe there exists a suitable privacy budget range (particularly $100 \leq \epsilon \leq 200$ in our evaluation) providing a good trade-off between utility and privacy protection. Our key observation drawn from our study is that differential privacy tends to perform better in collaborative inference for datasets with smaller intra-class variations, which, to our knowledge, is the first easy-to-adopt practical guideline.

Index Terms—Internet of Things, deep learning, collaborative inference, differential privacy, attack.

I. INTRODUCTION

Recent advancements in deep learning techniques have greatly empowered various Internet of Things (IoT) applications such as object recognition, human activity recognition,

J. Ryu, J. Kim, D. Won, and H. Kim are with the Department of Computer Science and Engineering, Sungkyunkwan University, South Korea (email: jhryu@security.re.kr, yaup22cc@likelion.org, dhwon@security.re.kr, hyoung@skku.edu).

Y. Zheng is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China (email: yifeng.zheng@hit.edu.cn).

Corresponding author: Yifeng Zheng

Y. Gao is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China (e-mail: yansong.gao@njust.edu.cn).

S. Abuadbba and S. Nepal are with CSIRO’s Data61 (email: Sharif.Abuadbba@data61.csiro.au, Surya.Nepal@data61.csiro.au)

Cong Wang is with the Department of Computer Science, City University of Hong Kong (email: congwang@cityu.edu.hk).

Fig. 1. Overview of collaborative inference.

health monitoring, and environmental sensing [1]–[6]. However, running a trained deep neural network (DNN) model for new inputs (i.e., DNN inference) would be resource-intensive and requires high-performance computing with high energy consumption, making it notably difficult to be directly deployed on IoT devices which are typically constrained in computing, memory, and energy resources [7], [8]. In addition, storage of DNN models requires a large memory volume, which can hardly be met by low-end IoT devices. An intuitive solution is to build DNN models on the cloud and transfer data from IoT devices to the cloud for conducting the inference. However, in such deployment, IoT devices’ data will be exposed to the cloud service provider, which may raise privacy concerns when IoT devices’ data contain sensitive or personal information.

Recently, the collaborative inference [9], [10] framework was introduced to obviate the need to expose such data from resource-constrained IoT devices while the DNN inference is still successfully executed. In this framework, a DNN model is split into a local part containing simple shallow layers of the DNN model—the shallow layers are to fit the resource-constrained IoT devices, and a remote part containing the remaining sophisticated layers. The local part is deployed on the resource-constrained IoT devices, while the remote part is deployed on the cloud, as illustrated in Fig. 1. The DNN inference for new inputs is then performed in a collaborative manner. In particular, given an input, the local device first passes it to the local-part model to produce an intermediate output. This intermediate output is sent to the cloud to continue forward inference computation over the remaining layers. Collaborative inference fundamentally eschews direct exposure of the raw input data to the cloud. In addition, it obviates the fetter of massive storage and computational overhead.
imposed by large DNN models on the resource-constrained IoT devices. Moreover, it alleviates concerns on the intellectual property of DNN models for the model provider compared with deploying the entire DNN models on IoT devices since the main sophisticated part of the model resides in the cloud.

While collaborative inference provides straightforward protection for the input data compared with an entirely cloud-based solution, the intermediate output values computed from the local-part model must be revealed to the cloud, which could still entail privacy risks \[11\], \[12\]. As a mitigation technique, Wang et al. \[11\] recently proposed a framework using differential privacy which aims to avoid the privacy leakage from the intermediate values. Differential privacy \[13\] has become the de facto privacy standard as it provides a rigorous mathematical framework for formalizing privacy guarantees, in terms of a parameter called privacy budget \(\epsilon\). The framework in \[11\] 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 takes advantage of a noisy training technique to endow the cloud-part model with robustness to perturbed data and alleviate the impact of noise perturbation on the inference accuracy.

Despite the valuable research efforts, this state-of-the-art differential privacy framework is mainly studied from the theoretical perspective. The practical usability of this framework in the presence of potent attack against collaborative inference remains unclear. In particular, in the presence of potent attack, the calibration of the privacy budget \(\epsilon\) as well as the corresponding trade-off between utility and actual privacy protection is not understood.

In light of the above, in this paper, we aim to shed some light on the following question:

**RQ:** Is it feasible to apply differential privacy-based perturbation framework to gain protection in collaborative inference while well retaining the utility in the presence of potent attack? And if so, are there any easy-to-observe conditions for assessment?

![Visual intra-class variation of each dataset](image)

Specifically, our main contribution lies in providing the first extensive evaluation of the state-of-the-art differential privacy framework for collaborative inference to understand the impact of different choices of the privacy budget \(\epsilon\) on both privacy and inference accuracy, in the presence of potent attack against collaborative inference. Recently, He et al. \[12\] proposed an input reconstruction attack targeting collaborative inference, which is designed to facilitate the cloud to recover the input data from the received intermediate output maliciously. Our key insight is to take advantage of this state-of-the-art attack to evaluate the empirical protection strength of the theoretical differential privacy framework for collaborative inference, with regard to varying settings of the privacy budget \(\epsilon\).

Our study aims to reveal practical insights for using the differential privacy framework to protect the input data privacy in the trending practice of collaborative inference, which, to our best knowledge, is the first study. In summary, this paper presents the following contributions:

- We propose to leverage the state-of-the-art input reconstruction attack to evaluate the practical usability of differential privacy for the demanding practice of collaborative inference for IoT applications.
- We conduct an extensive empirical study applying the attack against the differential privacy framework, over various datasets including SVHN, GTSRB, STL-10, and CIFAR-10 and with varying privacy budget \(\epsilon\) ranging from 0.1 to 5000. We evaluate and analyze the inference accuracy and protection efficacy in terms of both visualized assessment and quantitative metrics. To facilitate and inspire further investigation in this research line, we release our artifacts at this link\[1\].
- We draw practical insights about easy-to-observe empirical characteristics of datasets. Specifically, our empirical insight is that the practicality of achieving both privacy protection and maintaining accuracy is dataset dependent in the collaborative inference framework, and that a dataset with small visually perceived intra-class variation appears to achieve both properties simultaneously. More specifically, among the datasets used in our evaluation, the CIFAR-10 has the largest visually perceived intra-class variation, as demonstrated in Fig. 2. Accordingly, as shown in Fig. 3 the normalized accuracy drop on the CIFAR-10 dataset is the highest: 17.621%, given the largest tested privacy budget \(\epsilon\) that can still resist the reconstruction attack.

The remainder of this paper is organized as follows. Section \[II\] outlines preliminaries about the differential privacy-based collaborative inference framework and the input reconstruction attack. Section \[III\] presents comprehensive experimental evaluations. Section \[IV\] discusses our findings from our extensive evaluations and draw practical insights. Section \[V\] describes the related work. Section \[VI\] concludes this work.

## II. Preliminaries

### A. Collaborative Inference

The collaborative inference framework \[9\], \[10\] was put forward for IoT-cloud applications. As shown in Fig. 1 a trained DNN model, denoted by \(f_\theta\) and parameterized by model parameters \(\theta\), is split into two parts: a local-part model \(f_{\theta_1}\) and a remote-part model \(f_{\theta_2}\). 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

\[1\] It will be replaced with a working link upon paper acceptance.
x, the client first feeds x to the local-part model and obtains \( x^* = f_{\theta_2}(x) \), which represents an intermediate output. This intermediate output is then sent to the cloud, which further applies the remote-part model \( f_{\theta_1} \) to \( x^* \) and produces the ultimate inference result \( y = f_{\theta_1}(x^*) \).

### B. Differential Privacy

Differential privacy is a mathematical framework defined for privacy-preserving data analysis. The formal definition of \( \epsilon \)-differential privacy is as follows [13].

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

Intuitively, the above definition indicates that for any output in the range \( S \) of the mechanism \( M \), its probability of being produced from \( D \) is very close to that of being produced from \( D' \), as characterized by \( \epsilon \). That is, given any output, one can hardly tell whether it is produced from \( D \) or \( D' \). The parameter \( \epsilon \) is usually referred to as the privacy budget. A smaller \( \epsilon \) 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 [14]–[18]. A widely used probability distribution in differential privacy is the Laplace distribution, denoted by \( \text{Lap}(b) \), 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 [14], [19] for \( \epsilon \)-differential privacy works by sampling noises from \( \text{Lap}(b) \) and adding the noises to the output values of the function \( g(\cdot) \). Hence, to achieve \( \epsilon \)-differential privacy, \( b \) is set according to the global sensitivity \( \Delta g \) of the function \( g(\cdot) \), i.e., \( b = \frac{\Delta g}{\epsilon} \).

Let \( || \cdot ||_1 \) denote the \( l_1 \) norm. The global sensitivity of \( \Delta g \) is defined as:

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

### C. 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. [11]. At a high level, this framework is comprised of two modules: one module on the client side which performs the differential privacy noise-based perturbation in the inference phase; and the other module on the cloud side, 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 \( \tilde{x} \leftarrow f_{\theta_1}(x) \). Then, noises sampled from the Laplace distribution are added to a bounded version of \( \tilde{x} \), producing the noisy intermediate output \( x^* \), which is sent to the cloud for inference. Note that bounding each value in \( \tilde{x} \) is needed because it is hard to directly estimate the global sensitivity of \( f_{\theta_1}(x) \) 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 \( x \) by randomly setting a portion of elements in \( x \) to zeros, masking some parts of \( x \) that are deemed highly sensitive.

As performing the perturbation will obviously degrade the accuracy performance of the DNN model, the design in [11] constructively takes advantage of noisy training to fine-tune the cloud-part model \( f_{\theta_2}(\cdot) \). The main idea is to perform training on both plain representations and noisy counterparts for the cloud-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 to [11] for details on the algorithm for noisy training. Let \( f_{\theta_2}^*(\cdot) \) denote the fine-tuned cloud-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 \( f_{\theta_2}^*(x^*) \) to the client as the inference result.

### D. Reconstruction Attack against Collaborative Inference

In a very recent study [12], He et al. proposed a reconstruction attack which allows the cloud to reconstruct the input image given the intermediate output and the local-part model in the collaborative inference framework. 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

---

**Algorithm 1** Differential Privacy Scheme Using Local Perturbation

**Input:** Input data sample \( x \); Bound threshold \( B \); Privacy budget \( \epsilon \).

**Output:** Noisy intermediate output \( x^* \).

1: \( \tilde{x} \leftarrow f_{\theta_1}(x) \)
2: \( d = ||\tilde{x}||_\infty \)
3: \( \hat{x} \leftarrow \tilde{x}/\max(1, d) \)
4: \( x^* \leftarrow \hat{x} + \text{Lap}(2B/\epsilon) \cdot \mathbb{1} \) Sampling and adding noises element-wise.

---

![Fig. 3](image-url) Comparison of (normalized) accuracy drops of the datasets given the largest tested privacy budget that still provides protection against the reconstruction attack (\( \epsilon = 200 \) for SVHN and GTSRB, and \( \epsilon = 500 \) for STL-10 and CIFAR-10).

- **Datasets**
  - SVHN
  - GTSRB
  - STL-10
  - CIFAR-10

- **Accuracy Loss** (%)
  - 0.92
  - 4.719
  - 5.681
  - 17.621
Algorithm 2: Reconstruction Attack

**Input:** Local-part model \( f_{\theta_1} \); Intermediate output \( f_{\theta_1}(x_0) \) for input \( x_0 \); Maximum number of iterations \( T \); Hyperparameters \( \lambda \) for total variation and \( s \) for step size.

**Output:** Reconstructed \( \tilde{x} \) for \( x_0 \)

1. \( L(x) = \| f_{\theta_1}(x) - f_{\theta_1}(x_0) \|_2^2 + \lambda \cdot TV(x) \)
2. \( t = 0 \)
3. \( x(0) = \text{Init()} \)
4. While \( (t < T) \) do
5. \( x^{(t+1)} = x^{(t)} - s \cdot \frac{\partial L(x^{(t)})}{\partial x^{(t)}} \)
6. \( t = t + 1 \)
7. end
8. return \( \tilde{x} = x^{(T)} \)

### Table I
**Specifications of Datasets Used in Our Study**

| Dataset   | Training Set Size | Testing Set Size | Clipping Bound |
|-----------|-------------------|------------------|----------------|
| SVHN      | 73,200            | 26,000           | 7774.1494      |
| GTSRB     | 14,600            | 4,800            | 9613.522       |
| STL-10    | 10,000            | 3,000            | 8346.818       |
| CIFAR-10  | 50,000            | 10,000           | 10680.272      |

provides the local-part to the client. Algorithm 2 gives the details of the reconstruction attack.

Let \( x_0 \) denote an example input image and \( \tilde{x} \) 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 \( \tilde{x} \) to the local-part model \( f_{\theta_1} \) produces an intermediate output \( f_{\theta_1}(\tilde{x}) \) that is similar to the observed \( f_{\theta_1}(x_0) \). Here the similarity is measured by the Euclidean distance. Secondly, \( \tilde{x} \) 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 \( \tilde{x} \) is as piecewise smooth as possible.

### III. Comprehensive Evaluations

**A. Experimental Setup**

**Datasets.** We use four datasets in our comprehensive empirical evaluations, including SVHN, GTSRB, CIFAR-10, and STL-10. The overall specifications of these datasets are given in Table I. It is noted that for each dataset, the clipping bound as shown in Table II 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 detail 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 \((32, 32, 3)\).
- **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 6000 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.

**Neural network architecture.** The overall DNN architecture used in our evaluation is detailed in Table II which is the same as in [11]. The input size is \((32, 32, 3)\), and the number of output class is 10. Following the prior work [11], we first derive the model parameters of the local-part model 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 once it is well trained. The cloud-

---

2 Batch normalization is applied in our case to further improve the plain model accuracy.
part model is then trained 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 cloud-part model is the output obtained by feeding the data sample to the local-part model.

Hyperparameters. For each dataset, we use the ADAM optimizer for training of the cloud-part model, following [11]. The learning rate is set to $0.000001$ for SVHN, $0.0000002$ for GTSRB, $0.00000027$ for STL-10, and $0.000001$ 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. We vary the privacy budget $\epsilon$ 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.

Quantitative metrics. In addition to the visualization on reconstructed images, MSE, SSIM, and PSNR metrics are also adopted to quantify the reconstruction efficacy, which generally measure the difference between the original image and the reconstructed image. Let $A$ and $B$ denote the original image and reconstructed image respectively, with 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:

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

2) **Structural similarity** (SSIM) [24] 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 $\mu_A$ and $\mu_B$ are the mean value of pixels in image $A$ and $B$, $\sigma_A^2$ and $\sigma_B^2$ are the variances, and $\sigma_{AB}$ is the co-variance, 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 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:

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

B. Results over the SVHN Dataset

Utility under DP. The baseline model over the SVHN dataset without differential privacy (DP) achieves accuracy of $92.953\%$. In Fig. 4 we show the accuracy results of the DP method (Fig. 4 (a)) as well as the normalized accuracy loss (Fig. 4 (b)) against the baseline accuracy, under varying values of the privacy budget $\epsilon$. 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 is $88.686\%$ for $\epsilon = 5$ (normalized accuracy loss of $4.59\%$), $90.397\%$ for $\epsilon = 10$ (normalized accuracy loss of $2.74\%$), $92.083\%$ for $\epsilon = 100$ (normalized accuracy loss of $0.93\%$), and $92.249\%$ for $\epsilon = 1000$ (normalized accuracy loss of $0.75\%$). These results show that on the SVHN dataset the DP method can still achieve good accuracy highly close to the baseline accuracy under suitable $\epsilon$ values.

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 input reconstruction attack for some example testing images of the SVHN dataset. 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 $\epsilon$. As expected, the protection becomes less effective as the $\epsilon$ 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 the Sample 3 and Sample 4.

In Fig. 6 we show the evolution 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 $\epsilon$. 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 evolves, 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 $\epsilon$. No clear ascending trends can be observed with the increase of the privacy budget $\epsilon$. This suggests that PSNR is not an appropriate metric for measuring the resistance of the DP method against the attack in this context.
$\epsilon$ (From Strong to Weak Protection) | Original
--- | ---
0.1 | 0.2 | 0.5 | 1 | 2 | 5 | 10 | 20 | 50 | 100 | 200 | 500 | 1000 | 2000 | 5000
--- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | ---
Sample 1 | Sample 2 | Sample 3 | Sample 4 | Sample 5 | Original

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

![Graph](image1)

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

![Graph](image2)

Fig. 7. Impact of the privacy budget $\epsilon$ on accuracy (GTSRB).

Remark. 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 $\epsilon$ values (e.g., $5 \leq \epsilon \leq 200$), can achieve accuracy comparable to the baseline while providing resistance to the reconstruction attack.

C. Results over the GTSRB Dataset

Utility under DP. The baseline model over the GTSRB dataset without differential privacy (DP) achieves accuracy of 92.869%. Fig. 7 show the accuracy results of the DP method (Fig. 7(a)) as well as the normalized accuracy loss (Fig. 7(b)) with respect to the baseline accuracy, under varying privacy budget $\epsilon$. As depicted in the figure, the DNN model under the DP method has essentially no utility until $\epsilon$ exceeds 10. For $\epsilon \geq 10$, the accuracy achieved by the DP method is becoming close to the baseline accuracy. For instance, the accuracy is 66.8162% for $\epsilon = 10$ (normalized accuracy loss of 28.053%), 88.025% for $\epsilon = 100$ (normalized accuracy loss of 5.215%), and 89.5874% for $\epsilon = 1000$ (normalized accuracy loss of 3.533%). These results show that on the GTSRB dataset the accuracy loss due to the DP method is small under suitable $\epsilon$ values.

Protection efficacy. Fig. 8 shows from a visual perspective the protection levels of the DP method against the input reconstruction attack for some example testing images of the GTSRB dataset. As expected, the protection becomes less effective with the increase of the $\epsilon$ value. 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 the sample images can be (clearly) observed from the reconstructed images.

In Fig. 9 we show the evolution 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 $\epsilon$. 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...
privacy budget against the baseline accuracy, under varying values of the privacy budget \( \epsilon \). As depicted in the figure, the DNN model under the DP method has essentially no utility for \( \epsilon \) values \((\epsilon < 10\) for \( \epsilon \geq 10\), the accuracy achieved by the DP method is getting close to the baseline accuracy. For instance, the accuracy is 57.5934\% for \( \epsilon = 10\) (normalized accuracy loss of 14.464\%), 62.5984\% for \( \epsilon = 100\) (normalized accuracy loss of 7.031\%), and 62.6212\% for \( \epsilon = 1000\) (normalized accuracy loss of 6.997\%). These results show that on the STL-10 dataset, the DP method can achieve accuracy comparable to the baseline under suitable \( \epsilon \) values.

**Protection efficacy.** Fig. [11] shows some visual evaluation results regarding the protection levels of the DP method against the input reconstruction attack. As expected, the protection becomes less effective with the increase of the \( \epsilon \) 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 evolution 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 \( \epsilon \). 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 evolves, 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 \( \epsilon \).
E. Results over the CIFAR-10 Dataset

Utility under DP. The baseline model over the CIFAR-10 dataset without differential privacy (DP) achieves accuracy of 84.5%. We show in Fig. 13 the accuracy results of the DP method (Fig. 13(a)) as well as the normalized accuracy loss (Fig. 13(b)) against the baseline accuracy, with regard to varying privacy budget $\epsilon$. 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 69.755% (normalized accuracy loss of 17.449%) to 72.1144% (normalized accuracy loss of 14.657%), which is not close to the baseline accuracy of 84.5%. These results show that on the CIFAR-10 dataset, the DP method can retain meaningful utility of the DNN model yet the accuracy loss against the base accuracy is notable.

Protection efficacy. In Fig. 14, we show some visual results regarding the effectiveness of differential privacy against the input reconstruction attack, under varying privacy budget $\epsilon$. 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. Fig. 15 shows the evolution 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 $\epsilon$. 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 evolves, 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 $\epsilon$.

Remark. 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 $\epsilon$ values (e.g., $200 \leq \epsilon \leq 500$), can retain meaningful
and STL-10 datasets, the use of differential privacy is able
to achieve accuracy close to the non-private baseline. For
example, for $\epsilon = 200$ where privacy protection is ensured,
the (normalized) accuracy loss is 0.92\% on SVHN, 4.719\% on
GTSRB, and 7.566\% on STL-10 respectively, while it is
up to 17.449\% on CIFAR-10.

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

Our empirical insight is that differential privacy tends to
perform better for datasets with small intra-class variation in
collaborative inference, since according to our visual observation
CIFAR-10 has the largest intra-class variation among the
tested datasets. 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), the accuracy drop
due to differential privacy is 17.621\% for CIFAR-10, while
it is 5.681\% for STL-10, 4.719\% for GTSRB, and 0.92\%
for SVHN, respectively. 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.

**Remark on future direction.** We hope our initial study can

**IV. 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 can be used for collaborative inference in the presence of the reconstruction attack.** From our results
above, we consistently observe that the use of differential
privacy will not make the DNN model useless, and can provide
protection on the input privacy in collaborative inference.
For different datasets, however, our observation is that the
suitable intervals of the privacy budget $\epsilon$ 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 4.59\% while it is 74.43\% 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 $100 \leq \epsilon \leq 200$ provides 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
stimulate research activities for further in-depth investigation. For example, it would be interesting to explore ways to quantify intra-class variations and conduct a formal investigation regarding the interplay among intra-class variations of datasets, the sensitivity of datasets to the noise injected from differential privacy, and accuracy degradation.

V. RELATED WORK

Our work is related to prior works on evaluating the effectiveness of differential privacy in machine learning with attacks. In [27], Rahman et al. evaluate membership inference attacks against a differentially private DNN model which is proposed in [28]. In [29], 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 [30], 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 input reconstruction attack.

VI. CONCLUSION

In this paper, we initiate the first comprehensive study on assessment of the practical usability of differential privacy for collaborative inference in the presence of state-of-the-art input reconstruction attack targeting collaborative inference. We conduct an extensive empirical evaluation over up to four commonly used datasets, examining the impact of varying privacy budget $\epsilon$ on the aspects including inference accuracy, visual protection strengths, and quantitative metrics. Empirical insights and easy-to-adopt practical guidelines on the privacy-utility trade-offs have been drawn when differential privacy is deployed for collaborative inference in practice. We hope our work can lead to a deeper understanding of the practicality of using differential privacy for the protection of model input privacy in collaborative inference for IoT applications.

REFERENCES

[1] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. F. Abdelzaher, “Deepsense: A unified deep learning framework for time-series mobile sensing data processing,” in Proc. of WWW, 2017.
[2] V. Radu, C. Tong, S. Bhattacharya, N. D. Lane, C. Mascolo, M. K. Marina, and F. Kawsar, “Multimodal deep learning for activity and context recognition,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 1, no. 4, pp. 157:1–157:27, 2017.
[3] S. Yao, Y. Zhao, H. Shao, A. Zhang, C. Zhang, S. Li, and T. F. Abdelzaher, “Rdeepsense: Reliable deep mobile computing models with uncertainty estimations,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 1, no. 4, pp. 173:1–173:26, 2017.
[4] S. Yao, Y. Zhao, H. Shao, C. Zhang, A. Zhang, S. Hu, D. Liu, S. Liu, L. Su, and T. F. Abdelzaher, “Sensexp: Enabling deep learning for internet of things with a semi-supervised framework,” Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., vol. 2, no. 3, pp. 144:1–144:21, 2018.
[5] X. Liu, H. Li, G. Xu, S. Liu, Z. Liu, and R. Lu, “PADL: privacy-aware and asynchronous deep learning for iot applications,” IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6955–6969, 2020.
[6] Y. Li, H. Li, G. Xu, T. Xiang, X. Huang, and R. Lu, “Toward secure and privacy-preserving distributed deep learning in fog-cloud computing,” IEEE Internet of Things Journal, vol. 7, no. 12, pp. 11460–11472, 2020.
[7] S. Yao, Y. Zhao, A. Zhang, S. Hu, H. Shao, C. Zhang, L. Su, and T. Abdelzaher, “Deep learning for the internet of things,” Computer, vol. 51, no. 5, pp. 32–41, 2018.
[8] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. F. Abdelzaher, “Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices,” in Proc. of ACM SenSys, 2018.
[9] S. Teerapittayanon, B. McDanel, and H. T. Kang, “Distributed deep neural networks over the cloud, the edge and end devices,” in Proc. of IEEE ICDCS, 2017.
[10] J. H. Ko, T. Na, M. F. Amir, and S. Mukhopadhyay, “Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained internet-of-things platforms,” in Proc. of IEEE International Conference on Advanced Video and Signal Based Surveillance, 2018.
[11] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, “Not just privacy: Improving performance of private deep learning in mobile cloud,” in Proc. of KDD, 2018.
[12] Z. He, T. Zhang, and R. B. Lee, “Model inversion attacks against collaborative inference,” in Proc. of ACSAC, 2019.
[13] C. Dwork, “Differential privacy,” in Proc. of ICALP, 2006.
[14] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, “Calibrating noise to sensitivity in private data analysis,” in Proc. of TCC, 2006.
[15] C. Xu, J. Ren, L. She, Y. Zhang, Z. Qin, and K. Ren, “Edgesanitizer: Locally differentially private deep inference at the edge for mobile data analytics,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5140–5151, 2019.
[16] H. Duan, Y. Zheng, Y. Du, A. Zhou, C. Wang, and M. H. Au, “Aggregating crowd wisdom via blockchain: A private, correct, and robust realization,” in Proc. of IEEE PerCom, 2019.
[17] J. Ni, K. Zhang, K. N. Alharbi, X. Lin, N. Zhang, and X. S. Shen, “Differentially private smart metering with fault tolerance and range-based filtering,” IEEE Transactions on Smart Grid, vol. 8, no. 5, pp. 2483–2493, 2017.
[18] Q. Wang, X. Lu, Y. Zhang, Z. Wang, Z. Qin, and K. Ren, “Secweb: Privacy-preserving web browsing monitoring with w-event differential privacy,” in Proc. of SecureComm, 2016.
[19] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, “Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy,” IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 4, pp. 591–606, 2018.
[20] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, “Reading digits in natural images with unsupervised feature learning,” in Proc. of NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
[21] N. Stallkamp, M. Schlipping, J. Salmen, and C.igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” Neural Networks, vol. 32, pp. 323–332, 2012.
[22] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.
[23] A. Coates, A. Y. Ng, and H. Lee, “An analysis of single-layer networks in unsupervised feature learning,” in Proc. of AISTATS, 2011.
[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.
[25] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with convolutional neural networks,” in Advances in neural information processing systems, 2014, pp. 766–774.
[26] J. Liu, I. Chiotellis, R. Tribel, and D. Cremers, “Effective Version Space Reduction for Convolutional Neural Networks,” arXiv preprint arXiv:2006.12456, 2020.
[27] M. A. Rahman, T. Rahman, R. Laganère, and N. Mohammed, “Membership inference attack against differentially private deep learning model,” Transactions on Data Privacy, vol. 11, no. 1, pp. 61–79, 2018.
[28] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in Proc. of ACM CCS, 2016.
[29] B. Jayaraman and D. Evans, “Evaluating differentially private machine learning in practice,” in Proc. of USENIX Security, 2019.
[30] D. Bernau, P. Grassal, J. Roßl, and F. Kerschbaum, “Assessing differentially private deep learning with membership inference,” CoRR, vol. abs/1912.11328, 2019.