INDIVIDUAL PROPERTY INFERENCE OVER COLLABORATIVE LEARNING IN DEEP FEATURE SPACE

Haoxin Yang[1,2], *Yi Wang*[2*], *Bin Li*[1,3]

[1]Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen, China
[2]Dongguan University of Technology, Dongguan, China
[3]Shenzhen Institute of Articial Intelligence and Robotics for Society, Shenzhen, China
yhx1996@outlook.com, wangyi@dgut.edu.cn, libin@szu.edu.cn

ABSTRACT

Collaborative learning is used in multi-media applications to distribute computing tasks and data storage over multiple sites. Recent studies found that private data information can be derived from model updates between the server and clients. Yet, previous methods are limited by their capabilities of privacy inference in more general and practical situations. In this paper, we propose a novel property inference method in the deep feature space to overcome those limitations. In particular, our method can make inference decisions on the level of individual examples instead of a batch of examples. We can simultaneously perform multiple property inference attacks without the need of image reconstruction. The proposed method is evaluated on several image benchmark datasets, which demonstrates significant improvement of inference accuracy even in the presence of privacy protection schemes.

***Index Terms*—** Collaborative learning, privacy, property inference, gradient leakage

1. INTRODUCTION

With excessive data collection and extending collaborations, the conventional approach of centralized learning is facing bottleneck issues in data management, network communication, privacy protection and increasing demands for processing. To combat these problems, decentralized learning techniques are proposed as an alternative way to distribute computing tasks and data storage over multiple sides. Most notably is federated learning [1] which is a type of *collaborative learning* (CL) schemes that have been used in multimedia systems such as object detection [2], video analysis [3], and speech recognition [4]. In such context, CL is considered privacy protective as the private data are processed locally and do not leave the remote client devices.

However, recent studies showed that it is still possible to infer private information from model abstractions [5, 6, 7]. According to the goal of adversaries, there can be three types of privacy attacks in the context of CL [8]: 1) sample reconstruction, 2) membership inference, and 3) property inference. *Sample reconstruction* intends to reconstruct one or more training samples and/or their respective labels from a pre-trained model. *Membership inference* tries to determine whether sample x was part of a training set $\mathcal{D}$. *Property inference* extracts auxiliary information from the target model, e.g., the ratio of women and men or the age information in a patient dataset, when such information was not an encoded attribute or a label.

Yet, previous methods are limited by their capabilities of privacy inference in more general and practical situations. Sample reconstruction may be used as a stepping stone before mounting inference attacks by reconstructing the actual data sample [5, 9, 10, 11]. Recently, [5] proposes deep leakage from gradients (DLG) for image reconstruction based on gradient inversion. [9, 10, 11] improve DLG respectively by different gradient matching function, BatchNorm layer's statistics and a pretrained generator. However, these methods were limited by efficiency and generalization due to the complexity. There are score-based and direct-gradient-based methods that do not require image reconstruction to make inference but have their own limitations. [12] try to infer the input data information by the model output. However, the malicious user is allowed to manipulate the system by adapting model parameters and communication properties (a.k.a. *active attacks*). This is not applicable in our context of CL where adversaries is assumed *curious-but-honest* who can only observe but not interfere with the training process (a.k.a. *passive attacks*). [7] draws meta-characteristics of the training dataset from gradient updates of client model parameters without sample reconstruction. However, the attack can only decide whether a particular property occurs in a *mini-batch* data instead of judging for *individual* data samples.

Previous methods are either limited by their capacity of privacy inference in more general and practical situations or cannot make inference on individual examples. We aim to re-

*Corresponding Author

solve these problems in this paper. In particular, we note that reconstruction-based methods are developed often with an objective of improving the visual quality and image fidelity at the pixel level. Whereas this is not necessary for property inference tasks in most scenarios. Inspired from transfer learning, we propose a novel inference approach by reconstructing samples in the deep feature space that can take the advantages of both reconstruction-based and gradient-based methods.

Our main contributions are as follows.

- We demonstrate privacy leakage in the deep feature space and show that high-level feature learned for the model's main task also encode unintended information that is sufficient to make inference of private data.
- We propose a novel deep feature reconstruction method by using data-specific gradients in such a way that the reconstruction performance is not affected by the mini-batch size of client updates in CL.
- We design deep feature-based inference algorithms that perform property inference attack for individual sample. Performance evaluations show that our method can better cope with imbalanced property data in the client updates and cross-dataset inference.

2. THREAT MODEL

In this paper, we consider the general framework of CL by assuming K clients with a common learning task of image classification and training collaboratively for a shared model by synchronous SGD (s-SGD)[13] on the server S.

2.1. Attack Semantics

We follow the practice in [5] to allow only gradients and their updates sent to the server from local clients. The server may use a pre-trained model to initialize the shared model parameters $W^{(0)}$. At the t-th iteration, parameters of the shared model $W^{(t)}$ are downloaded from the server to clients. Each client then trains the model on a new batch sampled from its local dataset at the client device. Local gradient updates of the client model k, denoted by $\mathbf{g}_k^{(t)}$ for $k = 1, 2, ..., K$, are sent back to the server for updating the shared model by s-SGD:

$$W^{(t+1)} = W^{(t)} - \eta \sum_{k=1}^{K} \frac{m_k}{M} \cdot \mathbf{g}_k^{(t)} \qquad (1)$$

where m_k is the mini-batch size for each local update by client k and M is total size of the training data. η is the learning rate of CL.

2.2. Curious-but-honest Server

In this paper, we assume a malicious server who is *curious-but-honest*. That is, the server may derive client-private information without interfering the collaborative training nor affecting the model prediction performance. In addition to gradient updates from clients, the adversarial server may also exploit an auxiliary dataset $\mathcal{D}_{\texttt{aux}}$ commonly used for pre-training and quality assessments in the CL routines [14]. For the purpose of evaluations, the additional data must have the same distribution as the meta-data population.

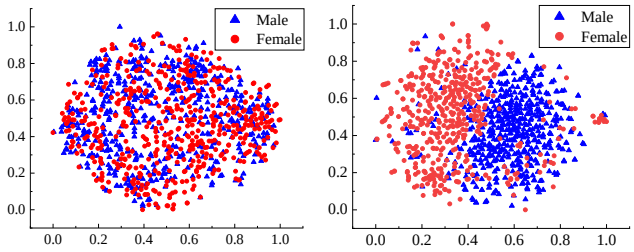

Fig. 1. Visualizations of 1,000 face images from the CelebA dataset [15] produced by t-SNE of deep features extracted from the last convolution layer of a shared model (a) at the beginning (i.e., $t = 0$) with random initialization, and (b) at the end (e.g., $t = 20$) of CL in the s-SGD setting.

3. PROPOSED METHOD

In this section, we first investigate privacy leakage in the deep space, i.e., whether the unintended information, e.g., gender or age group, is encoded in high-level feature representations learned for the main training task, e.g., face recognition, that is different from the inference task. Accordingly, we design a gradient inversion algorithm for deep feature reconstruction of individual examples. Then, we propose a unified framework that can construct several properties inference simultaneously in the deep feature space.

3.1. Privacy Leakage in Deep Space

In the study of transfer learning [16], it is well known that features learned on task A can be used for another task B to some extent, and that initializing a network with transferred features can improve generalization that lingers even after fine-tuning to the target dataset. This has motivated us to study privacy leakage in the deep space as high-level features may be exploited to perform a secondary tasks. To that end, we shall first corroborate that high-level feature representations do encode the unintended information unrelated to the main task of learning.

We design an experiment by training a CNN model of face recognition in the framework of CL. We use t-distributed Stochastic Neighbor Embedding (t-SNE) [17] to visualize any implicit data structure in the deep space, which converts the high-dimensional Euclidean distances between data points into conditional probabilities that present similarities. Fig. 1 (a) plots the t-SNE map for the random features when

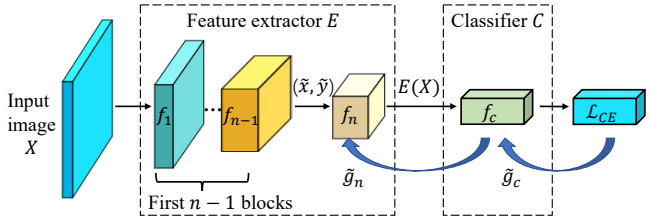

Fig. 2. Proposed deep feature reconstruction on CNN.

$t = 0$ at the beginning of CL. It can be seen that the data points are almost uniformly distributed in the manifold regardless of the gender class.

We then start the collaborative training of the shared CNN model on the CelebA dataset [15]. As the round of CL increases, the shared model is updated with actual information learned from more data samples. Fig. 1 (b) plots the t-SNE map for the deep features extracted from the last convolution layer of the CNN at the end of CL (i.e., t=20). It can be seen clearly that the data points are clustered with respect to the two gender classes even though the gender information is not provided to the CNN model during the entire process of CL. This is an inadvertent leak of data property information (e.g., gender in this case) in the deep space.

3.2. Deep Feature Reconstruction

In the previous section, we show that there is auxiliary information encoded in the deep features even if they are learned for another task independent of the unintended information. This enables us to make privacy inference for individual examples in the deep feature space and eliminates visual quality issues that affect inference performance at the pixel level.

Fig. 2 illustrates the proposed scheme of deep feature reconstruction. The CNN model in general can be divided into two parts. The first part is a feature extractor that learns the deep feature representation of an input image, denoted by $E(X)$, followed by a classifier C with the decision function f_c designed for the main task of CNN. Without loss of generality, we assume n blocks of convolution layers in E, denoted by $f_1, f_2, ..., f_n$, respectively. In particular, f_n produces the last-layer features $E(X)$.

Fig. 1 plots the t-SNE maps using $E(X)$, which shows that $E(X)$ can be used to make inference. In cases when the difference between the main and auxiliary tasks increases, one may resort to lower-layer features as they have more general information [16]. However, this tends to be more affected by the actual network architecture such as batch normalization, activation function, and etc. As more layers are involved, it will increase not only the time complexity but also reconstruction errors. In our experience, we found that the use of last-layer features $E(X)$ is sufficient for the property inferences over all experimented datasets presented in this paper.

Following the premises of CL, the server sends the shared model to each client at the beginning of each round CL, and updated by s-SGD using the client gradient updates $\mathbf{g}_k^{(t)}$ in multiple rounds of CL. For conciseness and the ease of presentation, we drop the subscript k and superscript (t) in $\mathbf{g}_k^{(t)}$ hereafter for a particular client update in a round. Thus, we have *data-specific gradients* $\mathbf{g} = [g_1, g_2, ..., g_n, g_c]$ for updating model parameters by (1) of the corresponding blocks, i.e., $f_1, f_2, ..., f_n, f_c$, in Fig. 2.

We propose to reconstruct the last-layer features $E(X)$ from $\mathbf{g}$ specific to the mini-batch data of a client. As shown in Fig. 2, we randomly initialize a pair of dummy data $(\tilde{x}, \tilde{y})$, where $\tilde{x}$ is the *dummy features* and $\tilde{y}$ is a dummy label. The batch size of $(\tilde{x}, \tilde{y})$ is the same as the mini-batch size of a client. We then inject $(\tilde{x}, \tilde{y})$ into a copy of the shared model to learn dummy gradients by backward propagation of f_c and f_n, respectively, i.e.,

$$\tilde{g}_c = \nabla_{f_c} \mathcal{L}_{\texttt{CE}} \left[f_c \left(f_n(\tilde{x}) \right), \tilde{y} \right] \tag{2}$$

and

$$\tilde{g}_n = \nabla_{f_n} \mathcal{L}_{\texttt{CE}} \left[f_c \left(f_n(\tilde{x}) \right), \tilde{y} \right] \ , \tag{3}$$

where $\mathcal{L}_{\texttt{CE}}$ is the cross-entropy loss function. We intend to match these dummy gradients to the data-specific gradients for actual shared model updates received from the client. Therefore, we design the objective function as

$$\mathcal{L} = \lambda \cdot d\left(g_n, \tilde{g}_n\right) + d\left(g_c, \tilde{g}_c\right) \tag{4}$$

where $\lambda = 0.1$ and the difference between the dummy gradients $\tilde{g}$ and data-specific gradients g is measured by

$$d(g, \tilde{g}) = \left(1 - \frac{\langle g, \tilde{g} \rangle}{||g|| \cdot ||\tilde{g}||}\right) + \left(1 - \exp\left(\frac{-||g - \tilde{g}||^2}{\sigma^2}\right)\right) \tag{5}$$

which contains two terms. The first term is a cosine similarity distance. The second term is a Gaussian kernel based function with $\sigma^2 = \texttt{Var}(g)$. The latter is introduced because $\tilde{g} \ll g$ in many cases and thus the normal l_2 distance is problematic especially at the early stages. We can obtain the best $(\tilde{x}^*, \tilde{y}^*)$ by minimizing Eq. (4) with

$$(\tilde{x}_{j+1}, \tilde{y}_{j+1}) = (\tilde{x}_j, \tilde{y}_j) - \alpha \cdot \nabla_{(\tilde{x}_j, \tilde{y}_j)} \mathcal{L}(\tilde{x}_j, \tilde{y}_j) \tag{6}$$

for a number of iterations. We empirically set the learning rate $\alpha = 0.1$. The last-layer deep features of every individual samples in the batch can be computed from the best dummy features as

$$E(X) := f_n(\tilde{x}^*) \tag{7}$$

Note that we use two data-specific gradients in the client update, i.e., g_c and g_n correspond to the classifier block f_c and the last convolution block f_n. This has enabled us to use both backward and forward propagations for estimating $E(X)$, as shown in Fig. 2. We shall demonstrate later in the Section 4.3 that the inclusion of g_n plays an important role in stabilizing the inference accuracy when increasing the batch size.

3.3. Deep Feature-based Property Inference Attacks

Assume that the adversary has the auxiliary dataset $\mathcal{D}_{\mathtt{aux}}$ and data-specific gradients $\mathbf{g}_k^{(t)}$ updated from client k using the mini batch of private dataset $\mathcal{D}_k$ in the t-th round of CL. The adversary can estimate the deep features of client mini-batch data from $\mathbf{g}_k^{(t)}$, i.e., $E(X)$ for $X \in \mathcal{D}_k$, as shown in the previous section. He can also compute the deep features of samples in $\mathcal{D}_{\mathtt{aux}}$, denoted by $E(Z)$ where $Z \in \mathcal{D}_{\mathtt{aux}}$. Given $\mathbf{g}_k^{(t)}$, the adversarial goal is to decide if any $X \in \mathcal{D}_k$ has the specific property $\mathtt{p}$. We can label $Z \in \mathcal{D}_{\mathtt{aux}}$ with $\mathtt{p}$ and use pairs of the supervised data $(E(Z), \mathtt{p})$ to train a property inference model $f_{\mathtt{p}}$. We then use $f_{\mathtt{p}}$ to predict the property label of $E(X)$ reconstructed from $\mathbf{g}_k^{(t)}$ for individual samples in the mini batch to perform property inference.

4. PERFORMANCE EVALUATIONS

Datasets. We use a number of image benchmark datasets for performance evaluations. All the images are cropped to remove background and resize to 64×64. The datasets are

1) The CelebA Dataset contains 202,599 face images of 10,177 identities and with 40 labelled attributes. We use the first 1000 identities containing 21,152 images.

2) Large-scale Attribute Dataset (LAD) [18] has 78,017 images of 5 super-classes with 359 labelled attributes. We use the superclass of vehicles (LAD-Vehicles) containing 13,290 images of 50 categories.

3) CUB-200-2011[19] has total 11,788 images of 200 categories birds with 312 labelled attributes.

4) Pubfig83[20] has 13,838 face images of 83 identities picking from Pubfig[21] dataset with 73 labelled attributes.

Models. We use ResNet-18 [22] for the shared model in the setting of CL. For the property inference model $f_{\mathtt{p}}$, we use a two-layer fully connected (FC) neural network with 1024 and 512 hidden units and drop-out layer.

Experimental Setting. Unless otherwise specified, the mini-batch size is set to 64 and the number of clients in CL is set to 5. An Adam optimizer is used for training the inference model $f_{\mathtt{p}}$ with a learning rate of 0.0001 for 50 epochs.

4.1. Property Inference

Comparing with Reconstruction Based Methods.

Our method is not limited to small mini-batch size for model updates, which is a significant advantage. Existing reconstruction-based methods, such as DLG[5], Inverting-Gradients (IG) [9] and GradInversion[10], in general require very small batch size in order to be functional. Fig. 3 shows some examples. When batchsize is set to 1, the tested methods are able to reconstruct the image example in an update with high fidelity. However, the reconstruction results become hardly visible when batchsize is increased to only 8,

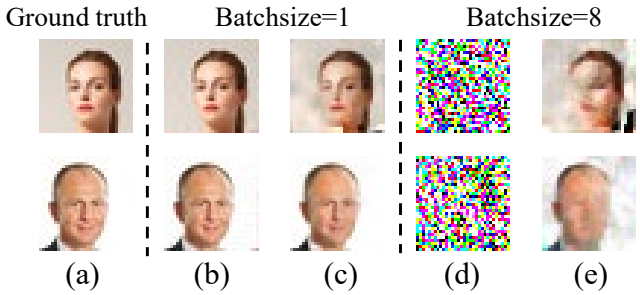

Fig. 3. Image reconstruction with different batchsize: (a) Ground truth from CelebA, (b) and (d) results by DLG[5], (c) and (e) results by GradInversion[10].

Table 1. The gender inference accuracy (in %) on CelebA at different mini-batch size for training updates.

Batchsize	DLG	IG	GradInversion	Proposed
1	93.12	94.86	94.79	**95.35**
8	50.00	67.34	52.34	**95.42**
32	50.00	54.31	73.23	**95.49**

which is considered too small for most machine learning models nowadays.

On the other hand, the proposed inference method reconstructs samples in the deep feature space and is not affected as much by the choice of mini-batch size. Table 1 shows the gender inference accuracy on CelebA. It can be seen that our feature-based method is able to retain the inference performance as batchsize increases. We also show the reconstructed performance under different batch sizes in Section 4.3.

Comparing with Non-Reconstruction Based Methods.

Table 2 presents the accuracy of inferring different data properties from client updates on CelebA, LAD-Vehicles and CUB-200-2011, respectively. Specifically, Unintended Leakage(UL)[7] trains a property inference model using gradient updates, while Honest-but-curious nets(HCN)[12] is a score-based method. The latter is an *active* attack by having interfered with the training process in order to encode auxiliary information in the output scores.

In all tested scenarios of Table 2, it can be seen that the proposed feature-based method is able to achieve a significant performance gain up to 14.5% and 7.7% comparing with UL and HCN, respectively. Note that our method is a *passive* attack that is in general more difficult and thus tends to have lower inference accuracy than an active attack such as HCN.

Imbalanced Property Data.

UL[7] can only test if data with a specific property occurs in a mini-batch but cannot decide which sample has that property, because the inference model is trained using the average gradient of mini-batch updates. This can be problematic when the number of data samples with the property is significantly decreased in the mini batch, a.k.a. *imbalanced property data*. Fig. 4 shows the effect by plotting the F1 score of inference

Table 2. Property inference accuracy (in %) by comparing with two non-reconstruction based methods.

Dataset	Property	UL	HCN	Proposed
CelebA	Gender	87.01	92.72	**95.75**
	Glasses	80.14	90.96	**94.64**
	Smile	77.53	85.54	**86.93**
	Young	65.83	73.47	**81.18**
LAD	Wheel	91.13	92.62	**96.13**
CUB	Beak	66.45	68.93	**75.08**

Table 3. Property inference accuracy using cross-datasets.

Method	Gender	Glasses	Smile	Young
UL	74.14	50.00	50.00	50.00
HCN	72.04	50.20	53.23	50.00
Proposed	**89.63**	**82.29**	**76.44**	**65.90**

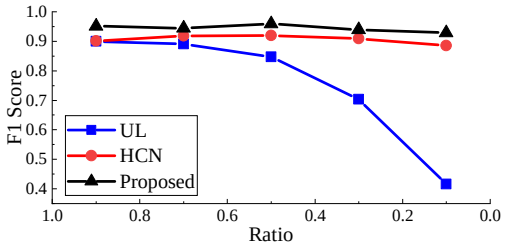

Fig. 4. Inference accuracy of gender w.r.t. the ratio of male vs. female in every client mini-batch data sampled from CelebA.

accuracy on CelebA w.r.t. the ratio of male vs. female samples in every mini-batch training data of clients sampled from CelebA. All three comparing methods perform similarly at the beginning when the ratio is 1:1. As the ratio of male samples decreases, the inference accuracy of [7] drops significantly to half whereas the proposed and the score-based methods retain performance even with imbalanced property data. Note that the latter two are both privacy attacks that can work on *individual* examples.

Cross-Dataset Inference.

Most existing methods, including [7] and [12], require an auxiliary dataset to learn a property inference model. Moreover, the auxiliary dataset must have supervised information of the main task as well as the same data distribution as the client training data. Whereas in our feature reconstruction method, we use the reconstructed deep feature to perform inference attack. Thus, we do not require accurate supervised information of the main task to build the property inference model. This gives our method great flexibility, especially in cross-dataset scenarios, e.g., when the auxiliary dataset $\mathcal{D}_{\text{aux}}$ is not available for training the inference model f_{p}.

Table 3 evaluates the inference accuracy using cross-datasets. Specifically, the property inference model is trained on Pubfig83 and then tested with gradient updates of local

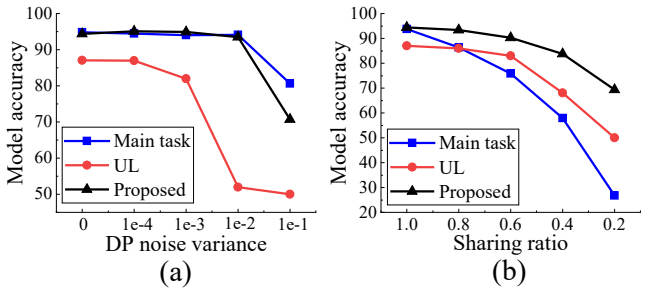

Fig. 5. Inference accuracy in the presence of (a) DP by varying noise, and (b) sharing less gradients in the updates.

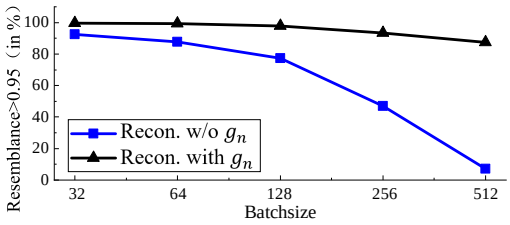

Fig. 6. Feature reconstruction by client gradient updates.

clients generated on CelebA. It can be seen that the proposed feature-based inference method is able to achieve significantly higher inference accuracy comparing with the other two methods. The gain of performance is at least 15% in all cases for cross-dataset inference in Table 3.

4.2. Inference Against Defence

We also evaluate the inference methods in the presence of two popular privacy protection schemes, namely differential privacy (DP) [23] and a privacy-aware technique by sharing less gradient (Sharing) in the setting of CL [24]. The amount of DP noise and ratio of sharing gradient relevant to the sensitivity. Thus, we can change the noise variance and the ratio of gradient sharing to control the trade-off between privacy and utility. In Fig. 5, we plot the model accuracy in terms of the *main task* for image classification in the presence of the two privacy protection schemes for reference.

Without loss of generality, we perform gender inference on CelebA dataset using the proposed method in comparison with the gradient-based scheme of [7]. As expected, the inference accuracy by both comparing methods decreases as the strength of privacy protection increases but that of our proposed method declines much slower. The gain of performance is up to 35% as shown in Fig. 5, where main task is the classification task of CL. This suggests that there is still much room for improvement of privacy defence.

4.3. Ablation Tests

We evaluate the performance of feature reconstruction in a batch by the percentage of samples that resemble the actual

client-specific data higher than 0.95 in terms of their cosine similarity in the deep feature space. Fig. 6 plots the results of ablation test by feature reconstruction with and without the data-specific gradient g_n that corresponds to the last convolution block f_n of the model shown in Fig. 2. As the batch size increases, we see that the performance drops quickly for reconstruction without g_n by more than 70% comparing with the proposed approach. This indicates that g_n is important by providing additional information for reconstruction especially with large batch size of updates.

5. CONCLUSION

In this paper, we propose a novel feature-based privacy inference method that can perform several property inferences simultaneously for individual sample in private client datasets through gradient updates in CL. The proposed approach is effective under different scenarios without the need of sample reconstruction at the pixel level. The inference performance retains even in the presence of two popular privacy protection schemes of collaborative training.

6. ACKNOWLEDGEMENT

The work was supported in part by NFSC(61876038, 61872244), Guangdong Basic and Applied Basic Research Foundation (Grant 2019B151502001), Shenzhen R&D Program(Grant JCYJ20200109105008228).

7. REFERENCES

[1] Jakub Konečnỳ, H Brendan McMahan, X Yu Felix, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," 2016.

[2] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang, "Fedvision: An online visual object detection platform powered by federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[3] He Li, Kaoru Ota, and Mianxiong Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE network*, 2018.

[4] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau, "Federated learning for keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[5] Ligeng Zhu, Zhijian Liu, and Song Han, "Deep leakage from gradients," *Advances in Neural Information Processing Systems*, 2019.

[6] Milad Nasr, Reza Shokri, and Amir Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *IEEE symposium on security and privacy*, 2019.

[7] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *IEEE Symposium on Security and Privacy*, 2019.

[8] Maria Rigaki and Sebastian Garcia, "A survey of privacy attacks in machine learning," 2020.

[9] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?," 2020.

[10] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Computer Vision and Pattern Recognition*, 2021.

[11] Jinwoo Jeon, Jaechang Kim, Kangwook Lee, Sewoong Oh, and Jungseul Ok, "Gradient inversion with generative image prior," in *Advances in Neural Information Processing Systems*, 2021.

[12] Mohammad Malekzadeh, Anastasia Borovykh, and Deniz Gündüz, "Honest-but-curious nets: Sensitive attributes of private inputs can be secretly coded into the classifiers' outputs," *ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[13] Jeffrey Dean, Greg S Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V Le, Mark Z Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, et al., "Large scale distributed deep networks," in *Advances in neural information processing systems*, 2012.

[14] Lingchen Zhao, Qian Wang, Qin Zou, Yan Zhang, and Yanjiao Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Transactions on Information Forensics and Security*, 2019.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *International Conference on Computer Vision*, 2015.

[16] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, 2014.

[17] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, 2008.

[18] Bo Zhao, Yanwei Fu, Rui Liang, Jiahong Wu, Yonggang Wang, and Yizhou Wang, "A large-scale attribute dataset for zero-shot learning," in *Computer Vision and Pattern Recognition Workshops*, 2019.

[19] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., California Institute of Technology, 2011.

[20] Nicolas Pinto, Zak Stone, Todd Zickler, and David Cox, "Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook," in *Computer Vision and Pattern Recognition Workshops*, 2011.

[21] Neeraj Kumar, Alexander Berg, Peter Belhumeur, and Shree Nayar, "Attribute and simile classifiers for face verification," in *International Conference on Computer Vision*, 2009.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition*, 2016.

[23] Cynthia Dwork, Aaron Roth, et al., "The algorithmic foundations of differential privacy.," *Foundations and Trends in Theoretical Computer Science*, 2014.

[24] Reza Shokri and Vitaly Shmatikov, "Privacy-preserving deep learning," in *ACM SIGSAC conference on computer and communications security*, 2015.