Privacy Partition: A Privacy-Preserving Framework for Deep Neural Networks in Edge Networks

Jianfeng Chi*, Emmanuel Owusu†, Xuwang Yin*, Tong Yu†, William Chan‡,
Yiming Liu†, Haodong Liu†, Jiasen Chen†, Swee Sim†, Vibha Iyengar†,
Patrick Tague†, and Yuan Tian*
* *University of Virginia*
Email: {jc6ub, xy4cm, yuant}@virginia.edu
† *Carnegie Mellon University*
{eowusu, tague}@cmu.edu,
{tong.yu, yiming.liu, haodong.liu, jiasen.chen, swee.horng.sim, vibha.iyengar}@sv.cmu.edu
‡*Google Brain*
wchan212@gmail.com

***Abstract*—The rise of the Internet of Things (IoT) encourages an emerging computing paradigm – *edge computing* – which leverages innovations in "last mile" communications infrastructure to provide improved support for connected devices and improved quality of service guarantees for compute-intensive services such as autonomous driving. Moreover, many high-value edge computing applications benefit from a privacy-preserving integration of resource-constrained connected devices, privacy-sensitive data streams, and resource-intensive analytic techniques like deep learning.**

We propose a practical method for privacy-preservation in deep learning classification tasks based on bipartite topology threat modeling and an interactive adversarial deep network construction in the context of edge computing. We term this approach *Privacy Partition*. A bipartite topology consisting of a trusted local partition and untrusted remote partition provides an apt alternative to centralized and federated collaborative deep learning frameworks in the case of deployment contexts such as IoT smart spaces, where users would like to restrict access to high-resolution data streams due to privacy concerns but would still like to benefit from deep learning services as well as external computational resources such as public cloud computing.

***Keywords*-Deep Learning, Edge Computing, Data Privacy**

I. Introduction

Presently, deep learning has been shown to outperform other machine learning solutions in a wide variety of problem areas including computer vision and natural language processing [1], [2], [3]. Deep-learning-based models benefit from increased computing power as well as a proliferation of massive data sets and data-streaming sources.

Real-time machine learning applications are some of the most significant consumers of user data and potentially one of the most significant producers of global network traffic. These analytic techniques require low latency guarantees from the network in addition to large quantities of data sourced from a large number of individual contributors. However, when cloud computing is the sole option for mobile offloading, these data-intensive technologies place significant pressures on limited network infrastructure due to the large amount of real-time data transmission as well as significant pressure on resource-constrained mobile embedded data-sources due to the high energy-demand of constantly refreshing wireless signals [4]. Still, the challenge of high performance network and computation requirements and demand for large volumes of user data are justified by the benefits of high value deep learning applications (e.g., the real-time computer vision processing used in self-driving vehicles and virtual reality).

Edge Computing [5] has been proposed to address such issues in mobile and cloud computing by routing and processing data at the edge of the networks. It has been shown that with the paradigm of edge networks, computing resources may be significantly reduced in terms of energy consumption and network latency. Additionally, edge computing architectures provide a promising medium for new privacy-preservation mechanisms by avoiding the need to send raw user data to remote service providers – easing the privacy risk potential of information leakages to honest-but-curious service providers or eavesdroppers.

In this work, we present a practical method for privacy-preservation in deep learning classification tasks based on bipartite topology threat modeling and an interactive adversarial deep network construction in the context of edge computing. We term this approach *Privacy Partition*. This framework is based on a bipartite deep neural network topology consisting of a trusted local partition and untrusted remote partition. It provides an apt alternative to centralized and federated collaborative deep learning frameworks in the case of edge networks, where users would like to restrict access to high-resolution data streams due to privacy concerns but would still like to benefit from deep learning services and external computational resources such as remote cloud data centers. We show the feasibility of our approach experimentally – we find that by using the proposed interactive adversarial training framework, the capacity for an adversary with access to deep network intermediate states to learn privacy-sensitive inputs to the network can be significantly attenuated.

II. Related Work

Edge computing is a distributed computing paradigm where computation is mainly performed at the edge of the network [5]. "Edge" devices refer to distributed computing nodes and network elements that are deployed close to connected devices and cyber-physical systems that consume their services. The new edge computing paradigm meets the need for the realization of scalable distributed computing [6]. However, solution addressing the privacy risk potential of edge computing has not been well studied.

Deep learning is one of widely used machine learning models. Recently, questions regarding how best to protect data privacy within deep networks (including the user data that is processed both during model learning and model usage) has raised significant interest in the community of security and privacy [7], [8], [9].

This work considers deep learning services in the setting of edge computing [10], [11], [12]. We propose a new bipartite topology consisting of a trusted local partition and untrusted remote partition. The proposed framework is applicable to edge computing deployments and can attenuate the privacy leakage to an honest-but-curious service provider.

III. Methodology

Prior research has shown that inputs to a deep network can be recovered from the hidden layer activation in some deep neural networks architecture such as the Convolutional Neural Network (CNN). Similarly, we were able to replicate this finding during the course of developing the framework proposed here for the non-invertibility of deep networks. In many cases, it is possible to recover inputs from the hidden layer activation by training a mapping function. Given similar training data as the target deep network and the corresponding hidden layer activation of the model, an attacker can compute $f_{\theta_a} : \mathcal{H} \to \mathcal{X}$ such that f_{θ_a} can map the hidden layer activation to input data.

In service time, we propose to "split" the deep learning model layers and deploy the "shallow" layers on the local computing nodes (e.g., edge devices), deploying the "deeper" layers of the deep network layers on the remote computing context (e.g., cloud servers). We term these partitions "*local layers*" and "*remote layers*".

The intuition behind the splitting of the deep network topology is that the data transformations, such as those resulting from the application of activation functions and pooling layers, exhibit some similarities to one-way functions – the resulting data representations are simultaneously better suited for the forward propagation, to successive network layers, of the features most salient to improving classification accuracy, while less applicable to backward propagation operations that may be used to recover the network inputs generated by any previous layers. We leverage and strengthen this type of information filtering, that occurs during the forward propagation of input data through a deep network, to lessen the ability of attackers to recover the often privacy-sensitive inputs to deep neural networks without incurring significant reductions to classification accuracy.

In effect, we would like to lessen the ability to recover network inputs by strengthening the invertibility of the local layer operations. To achieve this, we introduce an additional component during the model learning phase: *defender* (Θ_d). The role of defender Θ_d is to "mimic" the behavior of an attacker, which means, the defender attempts to recover the inputs given hidden layer activations. Then defender network Θ_d and the primary deep network Θ are trained concurrently with the defender network providing feedback regarding how well a potential attacker can recover inputs given hidden layer activations. The primary network responds in turn by iteratively optimizing itself to reduce the defenders recovery accuracy.

Recall that our primary goal is to learn a deep learning model $f_\theta = \mathcal{X} \to \mathcal{Y}$ with a sequence of training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$. According to our bipartite design, the deep learning model is formulated as $f_\theta = f_{\theta_l} \circ f_{\theta_r} = f_{\theta_r}(f_{\theta_l}(\cdot))$, where $f_{\theta_l} : \mathcal{X} \to \mathcal{H}$ is the function mapping input domain $\mathcal{X}$ to the domain of intermediate layer activations $\mathcal{H}$ in local partition Θ_l and $f_{\theta_r} = \mathcal{H} \to \mathcal{Y}$ is the function mapping $\mathcal{H}$ to output domain $\mathcal{Y}$ in remote partition Θ_r.

Within this framework, the defender learns mapping function $f_{\theta_d} = \mathcal{H} \to \mathcal{X}$. The objective function of the defender is formulated as follows:

$$\max_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} s\big(x_i, f_{\theta_d}(f_{\theta_l}(x_i))\big) \tag{1}$$

where $s(\cdots)$ is the similarity metric between the original input and the recovered input.

During the model learning phase, the primary model $f_\theta = f_{\theta_l} \circ f_{\theta_r}$ uses the recovery performance of defender model $f_{\theta_d} = \mathcal{H} \to \mathcal{X}$ as supplemental information for optimizing its parameters, making it harder for an attacker to recover input images at model usage time. This network capacity is formulated as follows:

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} l(y_i, f_\theta(x_i)) + \lambda \cdot s\big(x_i, f_{\theta_d}(f_{\theta_l}(x_i))\big) \tag{2}$$

where $\theta = \{\theta_l, \theta_r\}$, $l(\cdots)$ denotes the loss function for classification, and λ denotes the defender weight.

When the model is properly deployed to proximate edge computing infrastructure and remote public cloud infrastructure according to the bipartite topological design (i.e., during the model usage phase), the adversary wants to learn the best mapping function $f_{\theta_a} = \mathcal{H} \rightarrow \mathcal{X}$. The objective function of the attacker is formulated as:

$$\max_{\theta_a \in \{\theta_{a_1}, \ldots, \theta_{a_k}\}} \max_{\theta_a} \frac{1}{n} \sum_{i=1}^{n} s\big(\hat{x_i}, f_{\theta_a}(f_{\theta_l}(\hat{x_i})\big) \qquad (3)$$

The goal of the attacker is to recover the inputs of the model f_θ using the best available training data set $\mathcal{D}' = \{\hat{x_i}\}_{i=1}^{n}$ such that whenever intermediate network states $f_{\theta_l}(\hat{x_i})$ (corresponding to new data from the edge devices) are intercepted, the adversary can use f_{θ_a} to recover the input.

Note that the data set $\mathcal{D}'$ that the attacker obtains may differ from the model's training data set. In practice, the attacker's data set may be a similar data set that the attacker collects independently, or in the worst case, the same training data set used to learn network f_θ, or some admixture of the two.

In practice, the attacker may not know the precise network topology Θ used by $f_\theta = f_{\theta_l} \circ f_{\theta_r}$ (although the task type of the network is readily available and this information alone provides some general insights into the topology that is being used). In this case, the attacker may derive several estimates $\theta_a \in \{\theta_{a_1}, \ldots, \theta_{a_k}\}$, selecting the best match among the deep network architectures they devise based on the apparent quality of the recovered data.

IV. Conclusion

In this work, we propose *Privacy Partition* as a practical framework for reducing the privacy risk potential of an adversary with access to intermediate activation, or a significant portion of a deep network topology, conducting successful input recovery attacks. In essence, a deep network privacy partition lessens the ability to recover network inputs from intermediate network states by lessening the invertibility of local layer operations.

Future work will explore the feasibility of deploying this framework to large scale machine learning systems with integration of IoT software and hardware and security modules used to secure a local domain. Additionally, future work will investigate formal privacy guarantees and optimal invertibility conditions based on the proposed interactive adversarial training method.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[4] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46–54, 2016.

[5] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.

[6] K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," *Open Journal of Cloud Computing (OJCC)*, vol. 2, no. 1, pp. 16–24, 2015.

[7] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. ACM, 2015, pp. 1310–1321.

[8] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[9] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Generation Computer Systems*, vol. 74, pp. 76–85, 2017.

[10] H. Li, K. Ota, and M. Dong, "Learning iot in edge: deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[11] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 249–261, 2018.

[12] D. Li, T. Salonidis, N. V. Desai, and M. C. Chuah, "Deepcham: Collaborative edge-mediated adaptive deep learning for mobile object recognition," in *Edge Computing (SEC), IEEE/ACM Symposium on*. IEEE, 2016, pp. 64–76.