Advanced Topics in Distributed Systems

Critical review of selected literature on privacy preserving distributed machine learning

Sana Imtiaz

April 7, 2020

1 Selected literature

1. Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. *Semi-supervised knowledge transfer for deep learning from private training data.* ICLR 2017
2. Mohassel, P., and Zhang, Y. *SecureML: A system for scalable privacy-preserving machine learning.* IEEE S&P 2017
3. Hunt, T., Song, C., Shokri, R., Shmatikov, V., and Witchel, E. *Chiron: Privacy-preserving machine learning as a service.* CoRR 2018

2 Justification for literature selection

Machine learning (ML) is widely used in practice to produce predictive models for applications in business, health care, recommendation services, threat analysis, and authentication technologies. These models are more accurate when trained on huge amount of data collected from different sources. However, the massive data collection raises privacy concerns. Moreover, an ML model may inadvertently and implicitly store some of its training data, which may reveal sensitive information about training data upon careful analysis of the ML model. Selected literature deals with the problem of distributed and privacy preserving ML and provides different approaches to address the problem. The first paper demonstrates a generally applicable multi-tier ML approach to providing strong privacy guarantees for training data and proposes a distributed learning algorithm. The second paper presents a system for scalable privacy-preserving ML using a multi-party computation (MPC) approach. The third paper offers another interesting approach, privacy-preserving ML-as-a-service, which leverages the use of private computation units to reveal neither the training algorithm nor the model structure to the user, hence providing only black-box access to the trained model.

3 Paper 1: Review

3.1 Significance

The authors have demonstrated *Private Aggregation of Teacher Ensembles (PATE)* – a generalised distributed approach to providing strong privacy guarantees for training data. The proposed approach makes use of "teacher" models trained on disjoint subsets of private data to train a student model. Since the teacher models are not exposed, adversarial attacks exploiting the weaknesses of ML models are ineffective for this approach even if the internal workings of the student model are compromised. Unlike most works of literature, this approach is generic and can be applied to any model, even the deep neural networks; and achieves state-of-the-art privacy/utility trade-offs when combined with semi-supervised learning.

3.2 Contributions

- The authors demonstrate a general ML strategy that provides differential privacy for training data independent of the learning algorithm used. Due to this characteristic, PATE has become quite popular approach for training datasets that have labelled non-sensitive data components.
- This work explores four different approaches for reducing the student's dependence on its teachers. It applies GANs to semi-supervised learning which greatly reduces the privacy loss by radically reducing the need for supervision.
- The authors present a new application of a moments accountant technique for improving the differential-privacy analysis of knowledge transfer, which allows the training of students with meaningful privacy bounds.
- The proposed framework is evaluated on standard ML datasets with realistic bounds on privacy properties. The trained classifiers achieve an $(\epsilon; \delta)$ differential-privacy bound of $(2.04; 10^{-5})$ for MNIST and $(8.19; 10^{-6})$ for SVHN, respectively with accuracy of 98.00% and 90.66%.

3.3 Solutions

The proposed approach trains multiple models on disjoint subsets of sensitive dataset. These models are used as "teachers" for a "student" model. The authors use differential privacy to limit the effect of any single sensitive data item on the student's learning. The student learns to predict an output chosen by noisy voting among all of the teachers – Laplacian noise is added to the aggregated votes for labels provided by teacher models. Student cannot directly access an individual teacher or the underlying data or parameters. The authors assume that the student also has access to additional unlabeled public or non-sensitive data, which helps in unsupervised learning to estimate a good prior for the distribution. The authors demonstrate a number of training approaches and conclude that semi-supervised learning with GANs performs best for training the student models.

3.4 Experimental quality

Apart from the assumption that unlabeled non-sensitive version of the training dataset is available, the demonstrated work has reasonable and realistic assumptions. The proposed technique has been tested on standard ML datasets like MNIST and SVHN using realistic privacy guarantees. The details of the trained deep neural network have been clearly specified. A comparison with similar works has been provided to show the improvements in learning. Moreover, the authors have quantified the utility and privacy of the semi-supervised students, which makes it easier to visualize the impact of using private versions of ML models. The authors also show visually how much noise should be added to the queries made on ML models depending on the desired accuracy of query result, which is rare to find in other works of literature.

4 Paper 2: Review

4.1 Significance

This paper uses cryptography to ensure privacy preservation guarantees. The authors present new and efficient protocols for privacy preserving machine learning for linear regression, logistic regression and neural network training using the stochastic gradient descent (SGD) method. The proposed protocols follow a two-server model where data owners distribute their private data among two non-colluding servers, such that each server has a part of each data point which alone does not give information about the complete data point. These servers train various models on the joint data using secure two-party computation (2PC). The authors have developed new techniques to support secure arithmetic operations on shared decimal numbers, and propose MPC-friendly alternatives to non-linear functions such as sigmoid and softmax. It is claimed that this work implements the first privacy preserving system for training neural networks.

Note: In the distributed computing paradigm and big data settings, assuming a two-server scenario with data split is compute intensive and seemingly unrealistic, even though this work provides a good step towards faster variants of multi-party computation. In my opinion, this approach may be scalable as compared to other variants of MPC in terms of supporting sufficiently large datasets, but it is not scalable in the traditional distributed computing paradigm.

4.2 Contributions

- The designed privacy preserving linear regression protocol is several orders of magnitude more efficient than the state of the art solutions for the same problem.
- The authors implement the first privacy preserving protocols for logistic regression and neural networks training with high efficiency. For example, on a dataset of size 60,000 with 784 features, the privacy preserving logistic regression has a total running time of 29s while the privacy-preserving protocol for training a neural network with 3 layers and 266 neurons runs in 21,000s.
- The computation time for arithmetic operations on shared decimal numbers is significantly reduced with minimal loss of accuracy on computed results.
- The authors have designed MPC-friendly versions of the activation functions for neural networks.

4.3 Solutions

The paper presents a server-aided setting where the clients outsource the computation to two untrusted but non-colluding servers S_0 and S_1, typically an evaluator and a cloud service provider. The protocols are divided into a data-independent offline phase and an online phase. The proposed techniques use an oblivious transfer (OT) protocol, where a sender S has two inputs x_0 and x_1, and a receiver R has a selection bit b and wants to obtain x_b without learning anything else or revealing b to S. The online phase trains the model given the data, while the offline phase consists mainly of multiplication triplet generation. The authors use OTs both as part of the offline protocol for generating multiplication triplets, and in the online phase for logistic regression and neural network training, in order to securely compute the activation functions. In the proposed protocols, all intermediate values are secret-shared between the two servers and garbled circuits are used for computing results on secret values. For the activation functions, the authors make use of mini-batching for calculating the stochastic gradient descent.

4.4 Experimental quality

The paper presents a privacy preserving machine learning system implemented in C++ based on the proposed protocols. The experiments detail the hardware and network used, and focus on the training time of the protocol. In general, all techniques relying on cryptography for security and privacy have large computational overhead. Hence, the focus of the presented experiments is on reducing the computation time. The authors present their results on MNIST dataset in appendix, with 92% accuracy in the best case with a **training time of** 3.35 **days for the offline phase and** 70.6 **minutes for the online phase** of their protocol in the best case.

5 Paper 3: Review

5.1 Significance

This paper presents *Chiron* – a system for privacy-preserving ML as a service. Chiron is implemented using Intel® SGX enclaves. It conceals the training data from the service operator, as well as the training algorithm and the model structure from the user. Chiron runs the standard ML training toolchain in an enclave, and the model-creation code from the service operator is executed in a sandbox to prevent data leakage. To support distributed computing, it executes multiple concurrent enclaves which exchange model parameters via a parameter server. The authors present their results on the CIFAR and ImageNet datasets. The results show that the training performance and accuracy of the resulting models is suitable for use as ML-as-a-service.

Note: Chiron requires a dedicated architecture and programming model.

5.2 Contributions

- Chiron enables data holders to train ML models on an outsourced service without revealing their data. Moreover, it is applicable to all types of ML models so that the service provider is free to design the learning algorithm and data transformations.

- To enforce data confidentiality, trust is placed on a third party provider, a sandbox based on SGX enclaves, which prevents data leakage to the service provider. Moreover, users can verify the validity of the trained models on a validation set.
- Performance is a key feature. Chiron places a model-independent ML toolchain – Theano framework and C compiler – inside the hardware-protected enclave but outside the sandbox to ensure optimal privacy and performance trade-off. For example, Chiron slows down ImageNetLite training by 16% but preserves the accuracy of the trained model.
- Distributed and concurrent model training is supported.

5.3 Solutions

The system works as follows: Service provider loads code in the sandbox and makes one or more training enclaves available to the user. Users get a public-private key pair to communicate with the enclaves. User connects to training enclaves and submits data. Service provider code examines data, generates a model specification (model architecture, loss function, optimization function, and training hyperparameters) and passes it to the ML toolchain. ML toolchain uses the specifications to generate model-training code. Service provider code transforms data and breaks it into batches for training. Model-training code is invoked for each batch, updating the model. After the model has been created, the user measures its test accuracy on a validation set and and proceeds to use the model.

5.4 Experimental quality

The system architecture and threat model are clearly explained in the paper. Moreover, code snippets for building the models are presented. The results are demonstrated on standard benchmarks like CIFAR and ImageNet. The authors also explain the limitations of their work, those that arise from making use of commercial hardware and software components, as well as from their assumptions. The hardware setup and neural network design used for experiments is well explained. For CIFAR, accuracy as high as 88% is attained with a training time of around 3.5 hours. For ImageNet, `Top1` accuracy is reported around 52.4% with a training time of around 39 hours.

6 General Conclusions

- The field of distributed privacy-preserving machine learning is nascent and has huge room for research.
- The tuning of privacy-preservation parameters often needs to be done on a hit-and-trial basis, which might not be ideal for big data scenarios. For example, Paper 1 used 250 partitions of data (determined by experiment) and fine turned different privacy settings for the best results quoted in the paper.
- Cryptography based solutions have huge impact on efficiency of the system though they provide very strong privacy guarantees.
- Privacy preserving ML as a service is a good solution at the cost of placing trust on a third-party. Moreover, the efficiency requirements need further investigation.
- Appropriate choice for privacy preserving method should be applied while considering: the availability of public and private versions of dataset, efficiency and accuracy constraints of the ML application; and the adversary model and trust assumptions in the ML application settings.