Efficient and Private Federated Learning using TEE

Fan Mo, Hamed Haddadi

Systems and Algorithms Laboratory, Imperial College London

Background

Federated learning enables collaborative training on edge devices while keeping sensitive personal data local to the participants [2]. However, federated learning techniques can potentially leak information via the gradients present in shared models [3]. Such privacy leakage can have serious security and privacy implications.

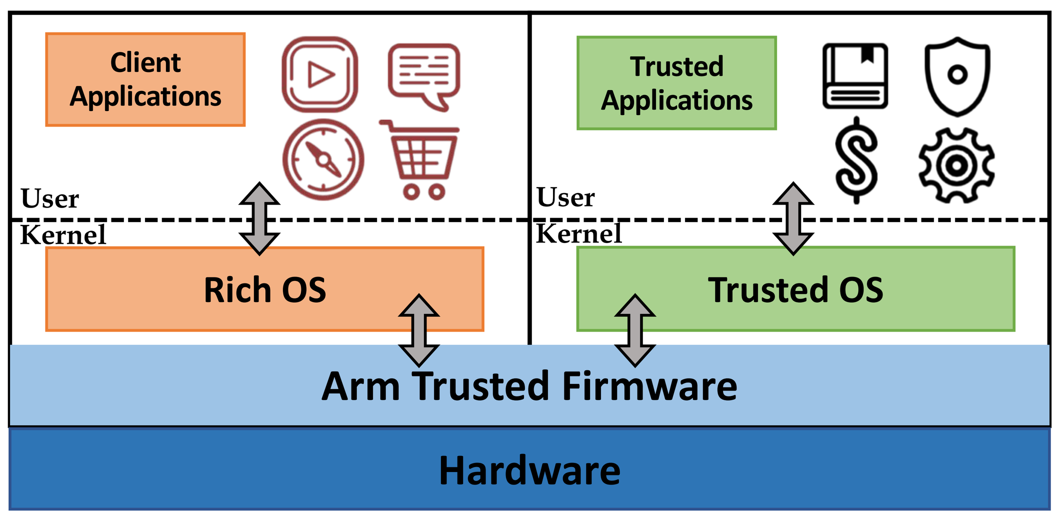

Figure 1:Simplified diagrammatic drawing of ARM TrustZone architecture

Leveraging the Trusted Execution Environment (TEE) implementation in ARM TrustZone (**Figure 1**), we focus on conducting private federated learning for edge computing without compromising accuracy and efficiency.

Proposed Framework

Partitioned Model Training

We present our framework that separates layers [5] and trains parts of the model in the TrustZone to prevent privacy leakage (**Figure 2**).

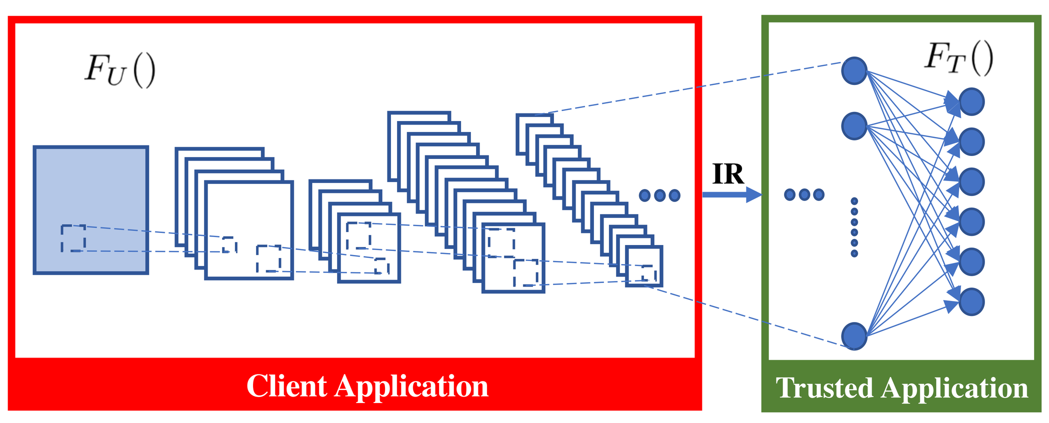

Figure 2:A partitioned DNN model in the TrustZone

Enhanced privacy-preserving techniques

- Data-oblivious trusted models [4]
  To defend side-channel attacks that listen at access patterns (e.g. following pseudo-code in ReLU activation) at layers in a DNN.
  if(input < 0) then:
  input = 0;
- Differential privacy-SGD [1]
  To obfuscate parameters and to guarantee privacy in untrusted parts.

Federated Learning with TEE

As an example, **Figure 3** shows the flow of model parameters during the training phase of Federated Learning.

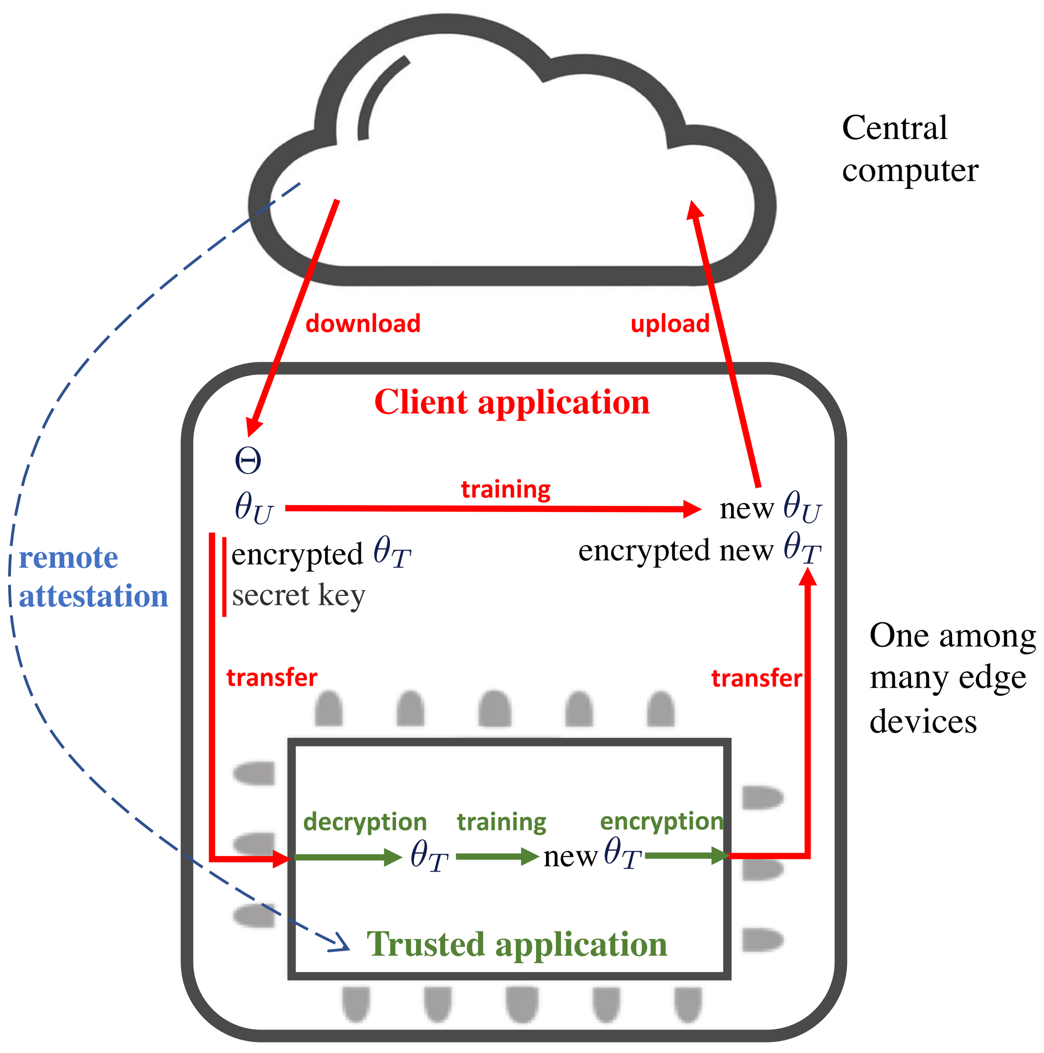

Figure 3:The transfer of model parameters during the partitioned federated learning

Experiment

- *MNIST* and *CIFAR-10* as the data sets
- *Open Portable TEE*, based on TrustZone, as the implementation
- *Darknet*, written in plain C language, as the DNN framework
- A *Raspberry Pi 3 Model B* as the setup
- *Le-net* for MNIST and a *Small-net* model for CIFAR-10 (**Figure 4**)

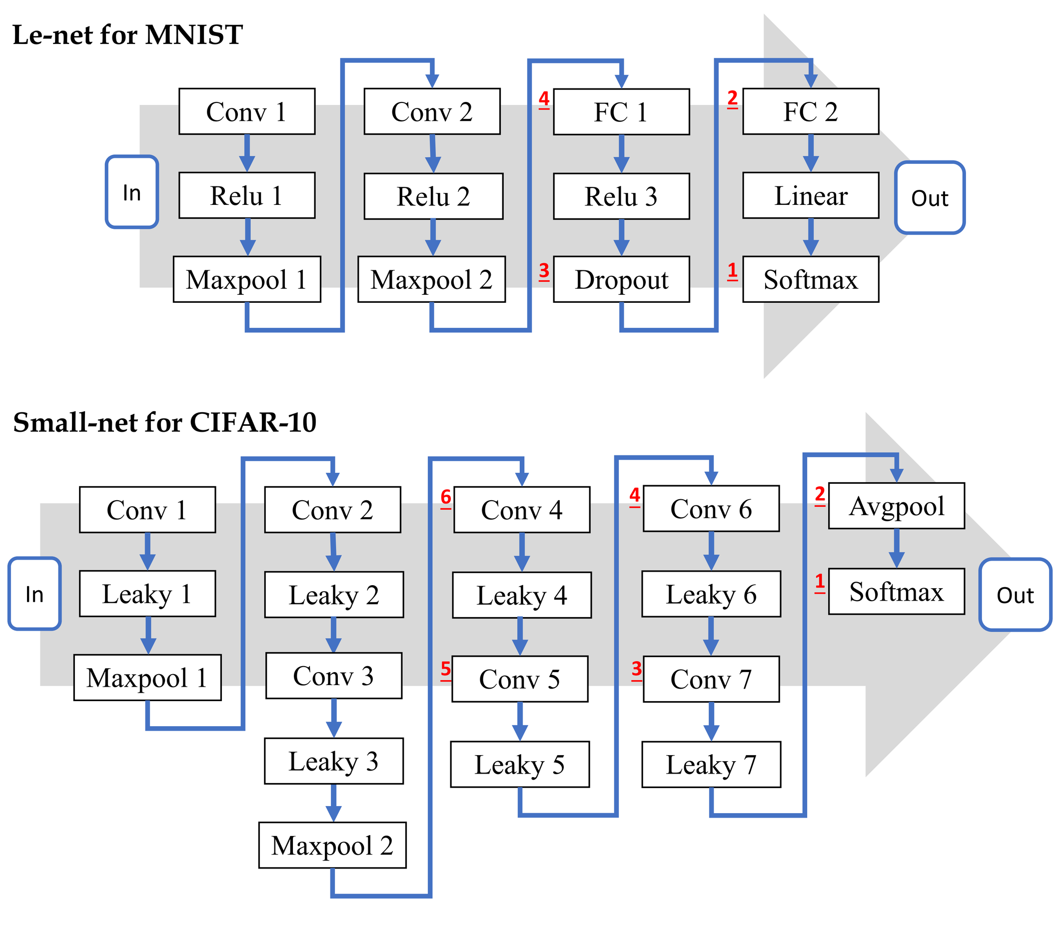

Figure 4:Partition of the Le-net model of MNIST and the Small-net model of CIFAR-10

Results

Overall, partitioning models does not significantly influence CPU usage (**Figure 5**). One exception is putting the maximum number of layers in TrustZone.

Partitioning models also slightly leads to a decrease of the CPU usage in the user mode, though consequently, it increases the CPU usage in the kernel mode.

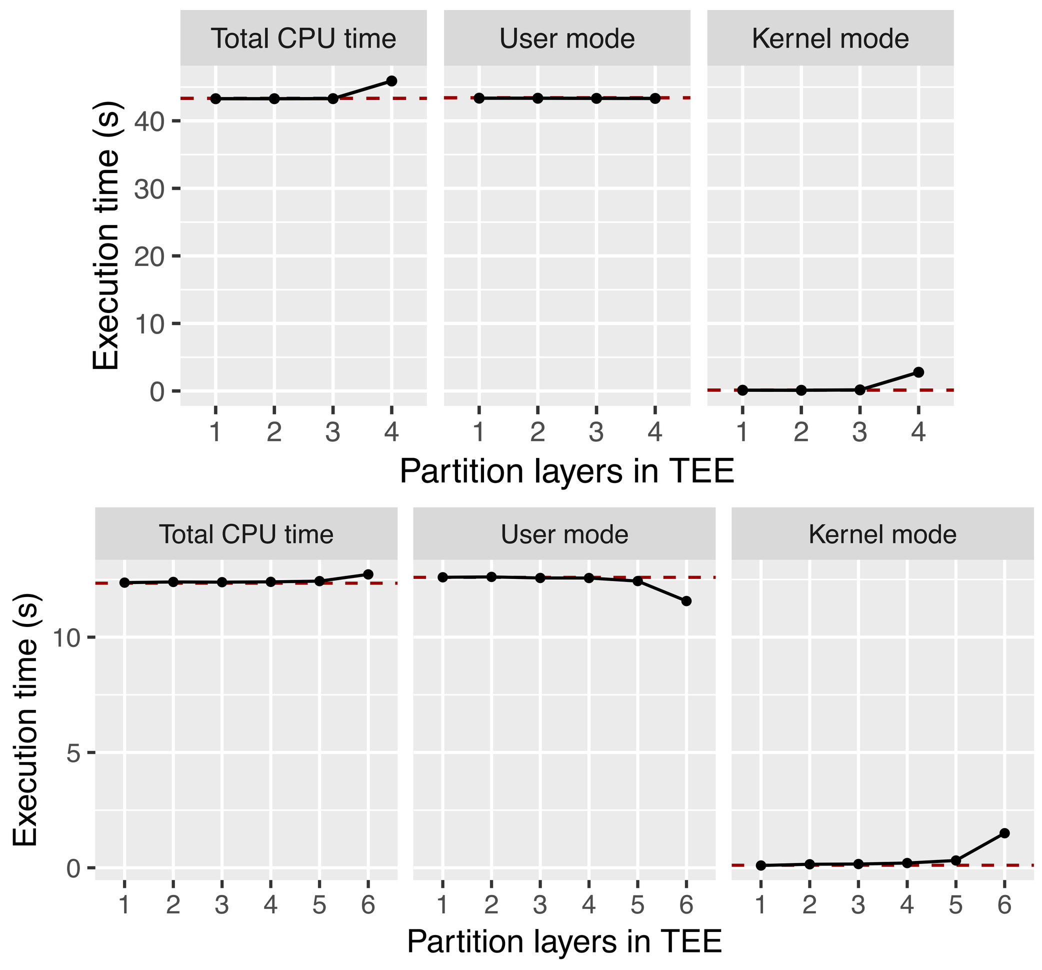

Figure 5:Execution time for partitioning models of MNIST (**top two figures**) and CIFAR-10 (**bottom two figures**)

The total cost of computation does not significantly increase (**Figure 6**).

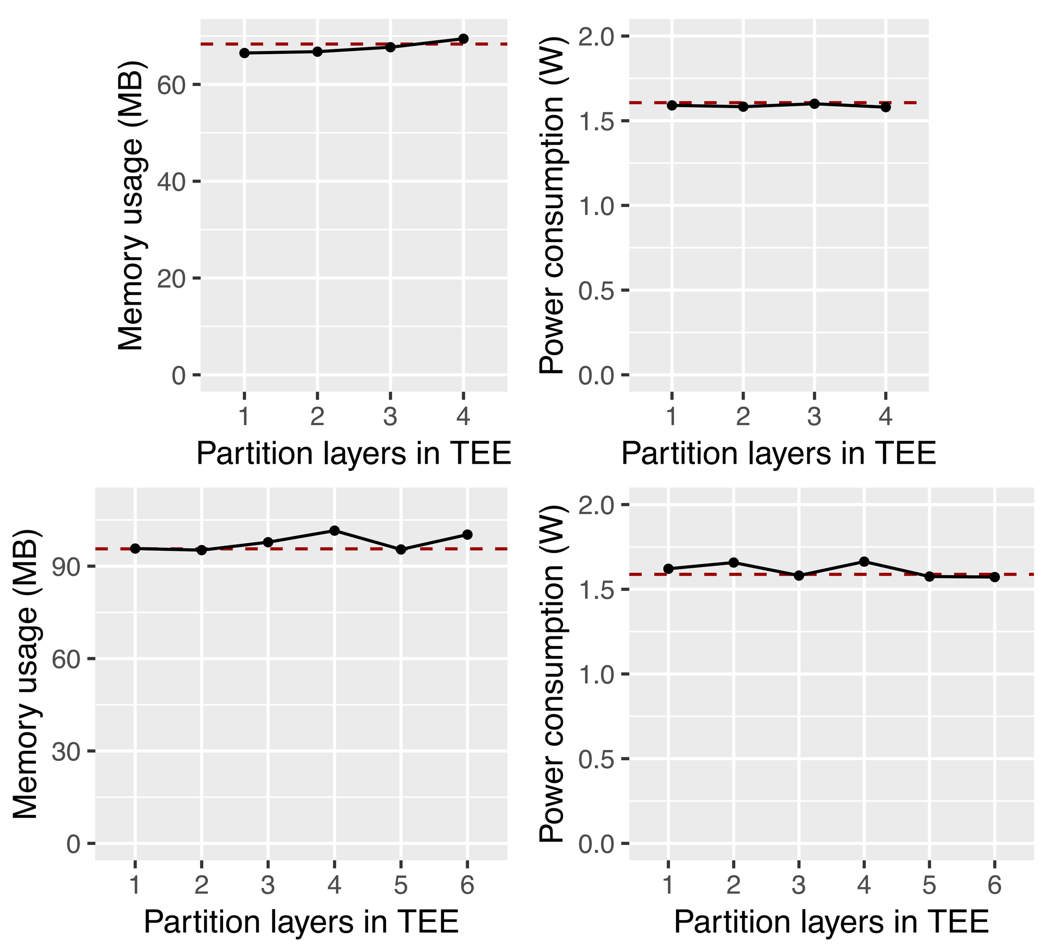

Figure 6:Memory usage and power consumption for partitioning models of MNIST (**top two figures**) and CIFAR-10 (**bottom two figures**)

Acknowledgements

EPSRC This work is partially supported by the EPSRC Databox grant (Ref: EP/N028260/1) and the EPSRC DADA grant (ref: EP/R03351X/1).

Contact Information

- f.mo18@imperial.ac.uk
- https://www.databoxproject.uk/

References

[1] M. Abadi, A. Chu, I. Goodfellow, and et al. Deep learning with differential privacy. 2016.

[2] B. McMahan and D. Ramage. Federated learning: Collaborative machine learning without centralized training data. 2017.

[3] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. 2019.

[4] O. Ohrimenko, F. Schuster, C. Fournet, and et al. Oblivious multi-party machine learning on trusted processors. 2016.

[5] S. A. Osia, A. Taheri, A. S. Shamsabadi, and et al. Deep private-feature extraction. 2019.