MPCLeague: Robust and Efficient Mixed-protocol Framework for 4-party Computation

Nishat Koti*, Arpita Patra* and Ajith Suresh*

*Indian Institute of Science, Bangalore

Secure Multi Party Computation (MPC)

- Introduced by Andrew Chi Chi Yao [1982]
- Enables **n** mutually distrusting parties to jointly compute a public function on their private inputs.
- Properties
  - Privacy: Nothing beyond function output is leaked
  - Correctness: All parties obtain the correct output of the function
- Adversarial model (Models the distrust among the parties)
  - Semi-honest: honest but curious
  - Malicious: arbitrarily deviate from protocol specification
- Corruption threshold
  - Honest Majority: majority are honest
  - Dishonest Majority: minority are honest
- Security levels
  - Security with Abort: honest parties may abort without receiving output
  - Fairness: either all parties or none get the output
  - Guaranteed Output Delivery (GOD): all parties guaranteed to obtain output

MPCLeague Protocol

- Model: 4PC, honest majority with security guarantee of GOD.
- Efficient and robust 4PC mixed protocol framework: at least 2x better than the state-of-the-art Trident [3].
- Efficient end-to-end conversions to switch between arithmetic/Boolean/garbled representations, efficient truncation, multi-input multiplication
- Two robust frameworks: MPCLeague1 that provides a fast online phase, and MPCLeague2 that provides best overall communication.

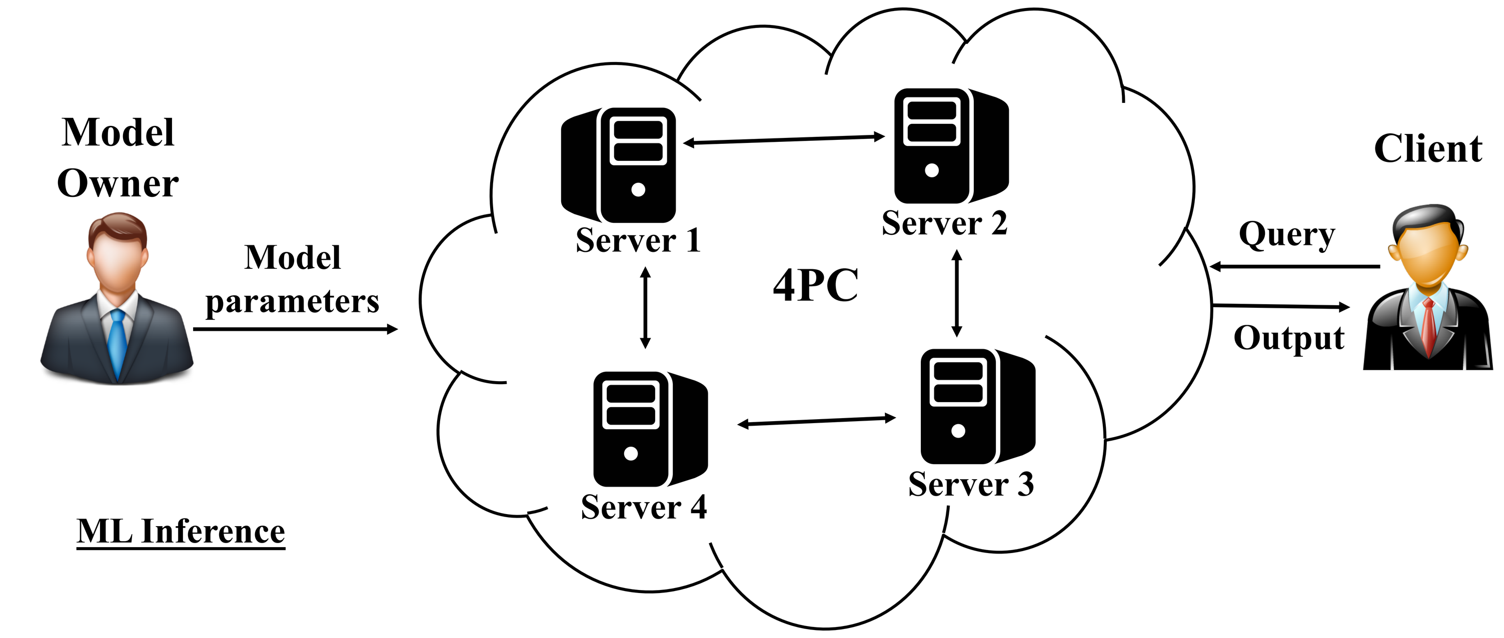
Ref	Training & Inference				Training	Inference	
	Com_{on}	Time_{on}	Com_{tot}	Time_{tot}	Cost	Cost	TP
Trident	○	◐	◐	○	◐	○	○
MPCLeague1	●	●	○	●	○	●	●
MPCLeague2	◐	○	●	◐	●	◐	◐

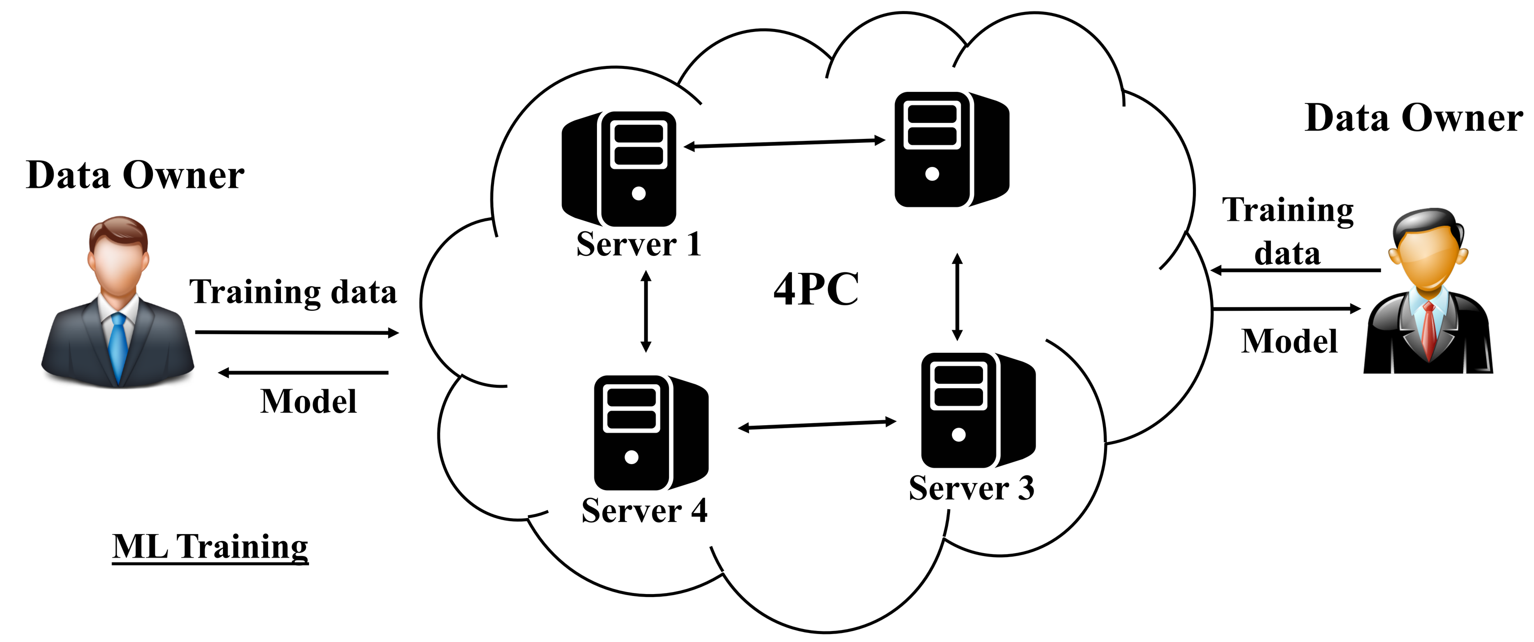
– 'Com' - Communication, 'Time' - Runtime, 'Cost' - Overall Monetary Cost, 'TP' - Online Throughput, on - online, tot - total (cf. Table 2)

– ● - best, ◐ - second best, ○ - least best protocol (w.r.t parameter considered).

Table 1: Comparison of Trident [14] with the two versions of MPCLeague for deep neural networks (cf. NN-4 in §6).

Privacy-Preserving Machine Learning (PPML)





- Outsourced server setting – computation outsourced to four untrusted but non-colluding servers.
- Works over ℓ-bit rings (64-bit for benchmarking).
- Follows the preprocessing paradigm.

Ref.	NN-1	NN-2	NN-3	NN-4
ABY3	318$	2530$	10200$	1350000$
Trident	28$	37$	175$	2880$
MPCLeague1	61$	70$	248$	3340$
MPCLeague2	26$	34$	160$	2520$

Table 4: Monetary Costs for Training (1 mini-batch, 10^3 iterations)

Ref.	SVM	NN-1	NN-2	NN-3	NN-4
ABY3	2750$	1100$	2610$	6220$	546000$
SWIFT (3PC)	919$	448$	451$	964$	3930$
SWIFT (4PC)	868$	418$	420$	902$	3390$
Trident	1150$	456$	457$	1130$	4340$
MPCLeague1	568$	324$	327$	688$	3110$
MPCLeague2	796$	347$	347$	828$	3290$

Table 5: Monetary Costs for Inference (10^5 predictions)

Results

- Neural Networks (NN)- Training and Inference
- Support Vector Machine (SVM): Inference only
- WAN- Google Cloud with 50Mbps, Implemented in C++17 using ENCRYPTO

Neural Network Architectures:

- NN-1: 3-layered fully connected network with ReLU activation after each layer (around 118K parameters).
- NN-2: Convolutional NN with 2 hidden layers, having 100 and 10 nodes
- NN-3: LeNet [1] has 2 convolutional layers and 2 fully connected layers with ReLU activation after each layer, additionally followed by maxpool for convolutional layers (around 431K parameters).
- NN-4: VGG16 [2] has 16 layers in total and comprises of fully-connected, convolutional, ReLU activation and maxpool layers (around 138 million parameters)

Benchmarking Parameters:

- Online and total protocol execution time.
- Online and total cumulative protocol execution time (sum of uptime of all),
- Online and total communication cost.
- Throughput for the case of inference.
- Monetary cost using Google Cloud pricing.

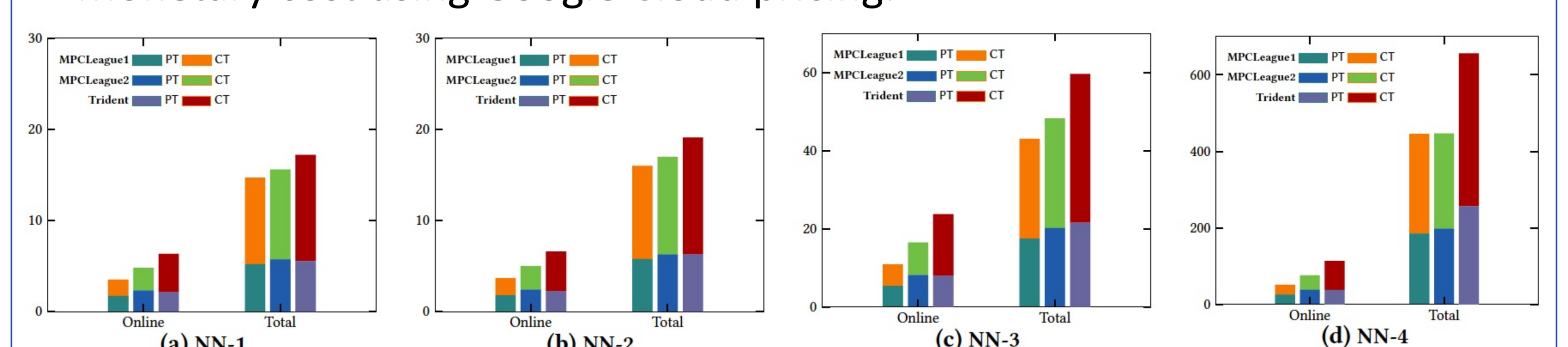
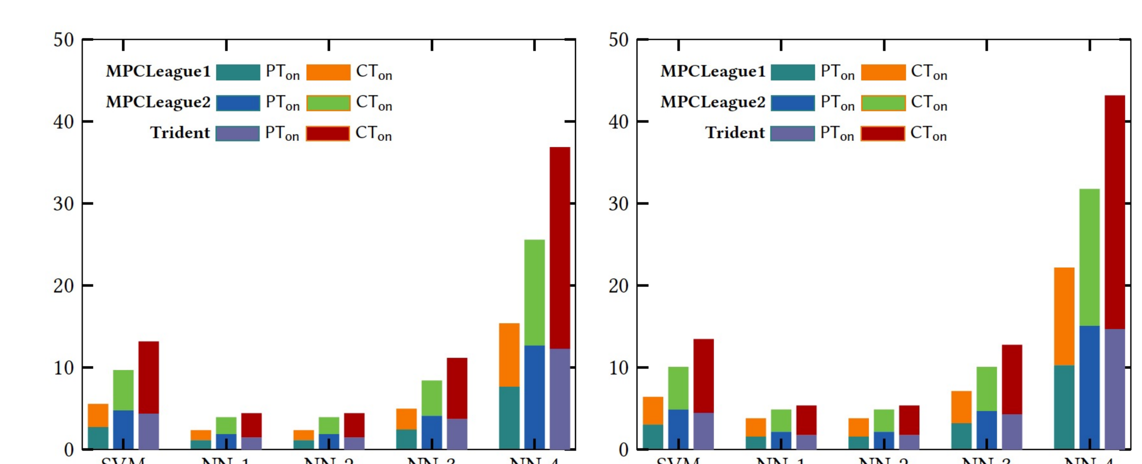


Figure 6: Training: in terms of PT_{on} and CT_{on} (cf. Table 2)



(a) Online RunTime **(b) Total RunTime**

Figure 7: Inference in terms of PT_{on} and CT_{on} (cf. Table 2)

Contact

Ajith Suresh and Nishat Koti

Indian Institute of Science, Bangalore

Email : {ajith, kotis}@iisc.ac.in

References

1. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient based learning applied to document recognition. Proc. IEEE (1998), 2278–2324.
2. Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
3. H. Chaudhari, R. Rachuri, A. Suresh. Trident: Efficient 4PC framework for privacy preserving machine learning. In NDSS 2020.
4. Nishat Koti, Mahak Pancholi, Arpita Patra, and Ajith Suresh. SWIFT: super-fast and robust privacy-preserving machine learning. To appear in USENIX 2021.
5. Payman Mohassel, and Peter Rindal. ABY3: A mixed protocol framework for machine learning. In CCS 2018.