Fusion: Efficient and Secure Inference Resilient to Malicious Servers

Caiqin Dong, Jian Weng, Jia-Nan Liu, Yue Zhang,
Yao Tong, Anjia Yang, Yudan Cheng, and Shun Hu
Outline

• Background
• Design Goals
• Our Solution: Fusion
• Performance
Machine Learning as a Service

Server

DL model
Motivation and Design Goals

Security Requirements

- Privacy Preservation
- Model Accuracy
- Computation Correctness

Client → Accurate inference results → Server

Malicious
Design Challenges

Possible Solutions

Model Accuracy
Zero-knowledge proof

Computation Correctness
Maliciously secure 2PC framework

Privacy

Need complex and careful design
An important observation

Client can know some computation results in advance
Our Key Insight

Mix-and-Check

1. Prepare public samples
2. Duplicate each query sample

Model Accuracy

Computation Correctness
Client Detects Server’s Malicious Behaviors

- Low-Quality model
- Incorrect computations for some samples

Inconsistent results
- same samples
- same samples

Inaccurate results
- public samples
Solution: **Fusion**

(I) Mixed Dataset Preparation

Mixed Dataset

- Randomly shuffled

Requested sample 1
Requested sample 2
Public samples

(II) Secure Inference Using Semi-Honest 2PC

Client

Server

DL model

Inference results

(III) Local Effective Checks

(a) Check model accuracy
(b) Check computation correctness
Solution: *Fusion*

- Client:
  - $R$: query samples
  - $T$: public samples

- Server:
  - A trained model
Client Prepares a Mixed Dataset

• (1) Prepare Mixed Dataset
Privacy-Preserving Inference Execution

- (2) Obtain Inference Results

Client → Semi-honest secure inference → Server

Inference results
Client Checks Inference Results

• (3) Simple-but-Effective Local Checks

Computation Correctness
Output consistency

Model accuracy
public samples
Optimal Number Selection

Given $R$, select appropriate $B, T$

Security Requirement
- Detect server's cheating

Cost Requirement
- Decrease the average cost
Security Requirement

Server succeeds in cheating

(2) Consistent-but-Incorrect

-(1) Model Accuracy

\[
\Pr[E_B] = \frac{\binom{R}{i}(iB)! (RB - iB)!}{(RB)!} \quad (2)
\]

\[
\Pr[E_T] = \frac{\binom{RB+T-iB}{T}}{\binom{RB+T}{T}} \quad (1)
\]
Client Selects Numbers Ensuring Security

Client

Search for the optimal numbers ensuring security

Security Requirements

\[ \Pr_{success} \leq 2^{-\lambda} \]

\[ \Pr_{success} = \Pr[E_T] \times \Pr[E_B] \]

Cost Optimization

\[ \text{arg min}_{B,T} \text{Cost}(B,T,R) = \frac{RB + T}{R} \]
Popular related works

- CCS17’ MiniONN (IPS compiler)
- Usenix18’ Gazelle
- CCS19’ LevioSA
- S&P20’ Cryptflow (TEE)
- S&P21’ SIRNN
- Usenix21’ ABY2.0

Threat Models
- Semi-Honest Security
- Malicious Security

Homomorphic Encryption + Garbled Circuits/Secret Sharing

Model Quality
Table I: Comparison between Cheetah-based Fusion and LevioSA (CCS19)

|      | Comm. (GiB) | Runtime (min) |
|------|-------------|---------------|
| LevioSA | 0.67       | 34.6          |
| Fusion  | 20.7        |               |

Runtime: $48.06 \times$ faster
Communication: $30.90 \times$ less

Table II: Performance of Fusion using different semi-honest inference protocols

| Number of Samples | ABY | DELPHI | CRYPTFLOW2 | Cheetah |
|-------------------|-----|--------|------------|---------|
| $10^{-3}$         |     |        |            |         |
| $10^{-1}$         |     |        |            |         |
| $10^{1}$          |     |        |            |         |
| $10^{3}$          |     |        |            |         |
| $10^{5}$          |     |        |            |         |
| $10^{7}$          |     |        |            |         |

(a) MNIST (LAN) (b) MNIST (WAN) (c) MNIST (WAN) (d) CIFAR-10 (LAN) (e) CIFAR-10 (LAN) (f) CIFAR-10 (WAN)
### Table III: Performance of Cheetah-based Fusion and comparison with semi-honest inference protocols

| Scheme | MNIST |  | CIFAR-10 |  |  |
|--------|-------|-------|----------|-------|-------|
|        | Comm. | LAN   | WAN      | Comm. | LAN   | WAN |
| Fusion | (2^3, 8, 100) | 24.499 | 487.500 | 850.000 | 30.080 | 575.000 | 975.000 |
|        | (2^5, 7, 100) | 12.102 | 228.125 | 425.000 | 14.838 | 284.375 | 481.250 |
|        | (2^7, 6, 100) | 8.106  | 154.688 | 283.594 | 9.949  | 189.844 | 323.438 |
|        | (2^9, 5, 100) | 6.210  | 118.164 | 215.820 | 7.617  | 145.508 | 246.289 |
|        | (2^13, 4, 100) | 4.799  | 90.686  | 166.968 | 5.886  | 112.341 | 191.724 |
|        | (2^19, 3, 100) | 3.580  | 68.204  | 125.570 | 4.407  | 84.297  | 143.249 |
| CRYPTOFlow2 [59] | 12.591 | 62.499 | 208.392 | 15.473 | 73.736 | 238.012 |
| DELPHI [46] | 123.412 | 563.924 | 1573.318 | 160.079 | 617.572 | 1814.989 |
| ABY [13] | 170.980 | 741.813 | 2293.557 | 207.421 | 850.366 | 2591.182 |

### Table IV: Performance on ResNet50

| Scheme | MNIST |  | CIFAR-10 |  |  |
|--------|-------|-------|----------|-------|-------|
|        | Comm. | LAN   | WAN      | Comm. | LAN   | WAN |
| Fusion | (2^3, 8, 100) | 39.921 | 20.410 | 34.241 |
|        | (2^5, 7, 100) | 19.714 | 10.082 | 16.912 |
|        | (2^7, 6, 100) | 13.205 | 6.750  | 11.326 |
|        | (2^9, 5, 100) | 10.117 | 5.173  | 8.678 |
| CRYPTOFlow2 [59] (SCI_{HE}) | 26.742 | 3.988 | 10.204 |
| CRYPTOFlow2 [59] (SCI_{OT}) | 281.497 | 4.795 | 39.466 |

2.64X slower 1.30X faster 1.18X faster
Conclusion

Strong Security

• Model accuracy
• Computation correctness
• Privacy preservation

High Efficiency

• Low average overhead
• Comparable efficiency with semi-honest protocols