FIFL: A Fair Incentive Mechanism for Federated Learning

Liang Gao, National University of Defense Technology
Li Li, State Key Laboratory of IoTSC, University of Macau
Yingwen Chen, National University of Defense Technology
Wenli Zheng, Shanghai Jiaotong University
ChengZhong Xu, State Key Laboratory of IoTSC, University of Macau
Ming Xu, National University of Defense Technology
Agenda

• Background and challenge
• The proposed FIFL mechanism
• Experiments and results
• Conclusion
Background

• Federated Learning, in which multiple devices collaboratively can train models by exchanging their model parameters instead of raw data.

• Crowdsourcing computing, the task publisher shares profit with workers to utilize workers’ data and computing resources.
Challenges

• How to accurately and efficiently identify the workers' utilities in the incentive mechanism?
• How to ensure fairness and reliability of the incentive mechanism under attacks and deceptions?
A case: Federated learning

Training process of servers

1.2. Upload local gradient slices

2.1 Collect gradient slices from workers

2.2 Calculate global gradient slices

Training process of workers

1.1. Local training with local data

1.4. Update model

2.3. Download global gradient slices

Workers

Local data

Gradient

Servers

1.3. Download global gradient slices

Worker-1

50th International Conference on Parallel Processing (ICPP)
August 9-12, 2021 in Virtual Chicago, IL
Mechanism

**Reputation module**
- The reputation of workers
  - The results from detection module
- Time
  - 1. Calculate reputation based on past behaviour
  - 2. Select high-reputation workers as servers

**Incentive module**
- Workers
  - Reputation
    - 0.8
    - 0.1
    - 1.0
- Servers
  - Contribution
    - 0.4
    - -1.2
    - 0.6
- Blockchain
  - Incentive
    - 1.4
    - -3.6
    - 1.6
- Smart contract
  - Incentive
    - 1.4
    - -3.6
    - 1.6

**Detection module**
- Local gradient
  - 1
  - 0
  - 1
  - 1
  - 1
  - 1
  - 0
- Global gradient
  - 1.0
  - 0.2
  - 1.0
- Time
  - The results from detection module
- 1. Detect attackers
- 2. Mark and eliminate attackers
- 3. Generate global gradient

**Contribution module**
- Similarity
  - 0.2
  - 1.5
- Threshold
  - 0.4
  - -1.2
  - 0.6
- The contribution of works
- 1. Calculate the similarity between local gradient and global gradient
- 2. Distinguish positive and negative contributions
- 3. Store messages about incentive, contribution and reputation in blockchains
- 4. Reward or punish workers according to their incentive

**Contribution module**
- Similarity distance
  - 0.2
  - 1.5
- Threshold
  - 0.4
  - -1.2
  - 0.6
- The contribution of works
- 1. Calculate the similarity between local gradient and global gradient
- 2. Distinguish positive and negative contributions
- 3. Store messages about incentive, contribution and reputation in blockchains
- 4. Reward or punish workers according to their incentive

---

50th International Conference on Parallel Processing (ICPP)
August 9-12, 2021 in Virtual Chicago, IL
Mechanism

Gradient

Attck detect

Normal or Abnormal

Update

Contribution and Reputation

Low-reputation

Attack

High-reputation

Local Gradient

Low-contribution

Negative contribution

High-quality local data

Worker 1

Low-quality local data

Worker N

Contribution at time $t_1$

Rewards of Worker 1

Contribution at time $t_4$
Experiment and results

Worker settings

• Honest workers
• Sign-flipping attackers
• Data-poison attackers
Experiment and results

Baselines

• Individual
• Equal
• Union
• Shapley
Experiment and results

Rewards distribution of workers

System revenue of different incentive mechanisms

Equal Individual Sharpley Union FIFL

Revenue as a % of FIFL

Equal Individual Sharpley Union FIFL

Incentive Mechanism

50th International Conference on Parallel Processing (ICPP)
August 9-12, 2021 in Virtual Chicago, IL
Experiment and results

Attackers’ damage

Effectiveness of FIFL in unreliable scenario
Experiment and results

Effectiveness of attack detection module

- No attack
- With attack detection, \( p_s = 10 \)
- Without attack detection, \( p_s = 10 \)

![Graph showing communication iteration and test loss with and without attack detection.](image-url)
Experiment and results

Effectiveness of reputation module

Effectiveness of contribution module

Set $p_d = 0.2$ as threshold
Experiment and results

Effectiveness of incentive module for data-poison attacker

![Graph (a)](image)

Set $p_d = 0.2$ as threshold

Effectiveness of incentive module for sign-flipping attacker

![Graph (b)](image)

Rewards

Communication Iteration

Punishment

Communication Iteration

Set $p_d = 0.2$ as threshold

$p_a = 0.0$

$p_a = 0.05$

$p_a = 0.1$

$p_a = 0.2$

$p_a = 0.3$

$p_a = 0.4$

$p_a = 0.5$

$p_a = 0.6$

$p_a = 0.7$

$p_a = 0.8$

$p_a = 0.9$

$p_a = 1.0$

$p_s = 2$

$p_s = 4$

$p_s = 6$

$p_s = 8$
Conclusion

• Attack resilience
• Fairness incentive
• Higher system revenue
Thanks for Listening!

• Liang Gao (gaoliang.gl@outlook.com)