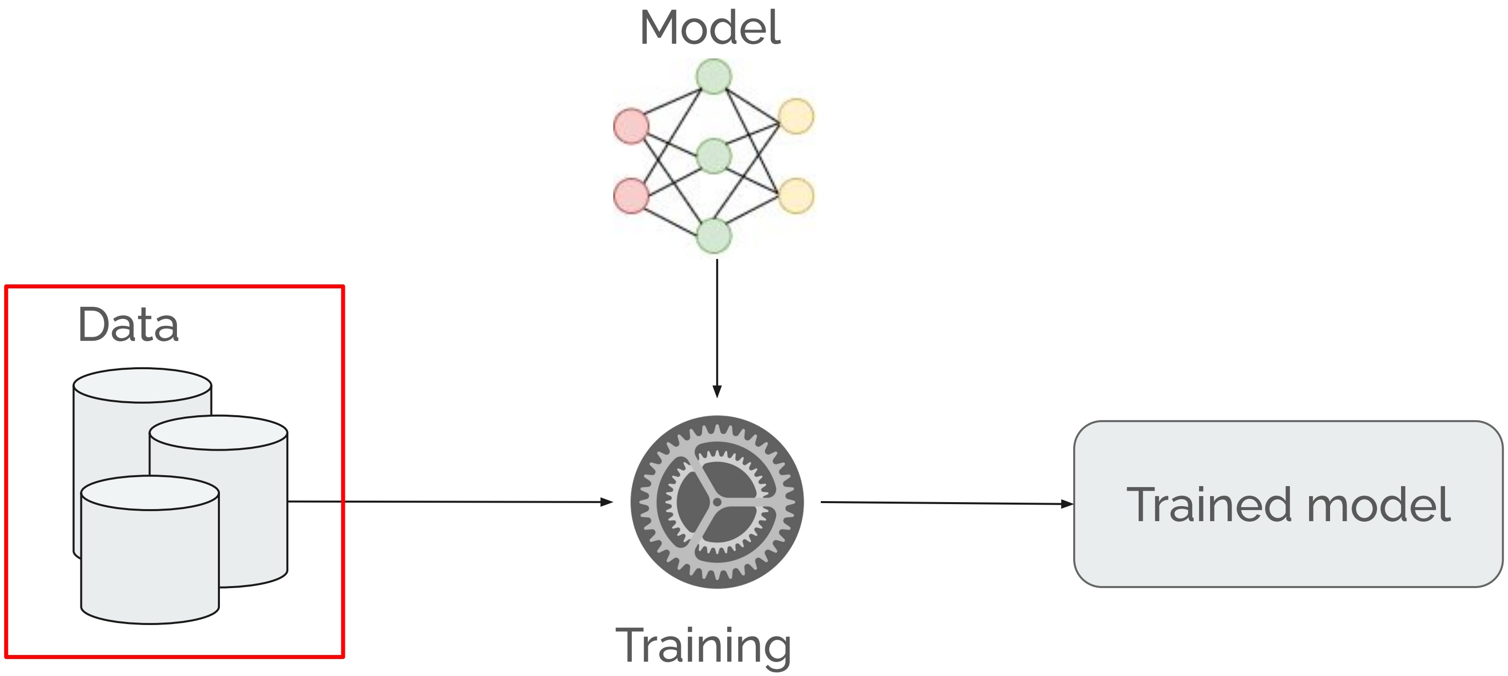
Privacy-Preserving Contribution in Federated Learning

- **Supervisor**: Andrea Vitaletti
- **Student**: Gabriele Lerani
- MSc Engineering in Computer Science

Outline

1. ML vs FML
2. Research questions
3. Zero-Knowledge proofs
4. Proposed framework
5. Experiments and results
6. Possible improvements

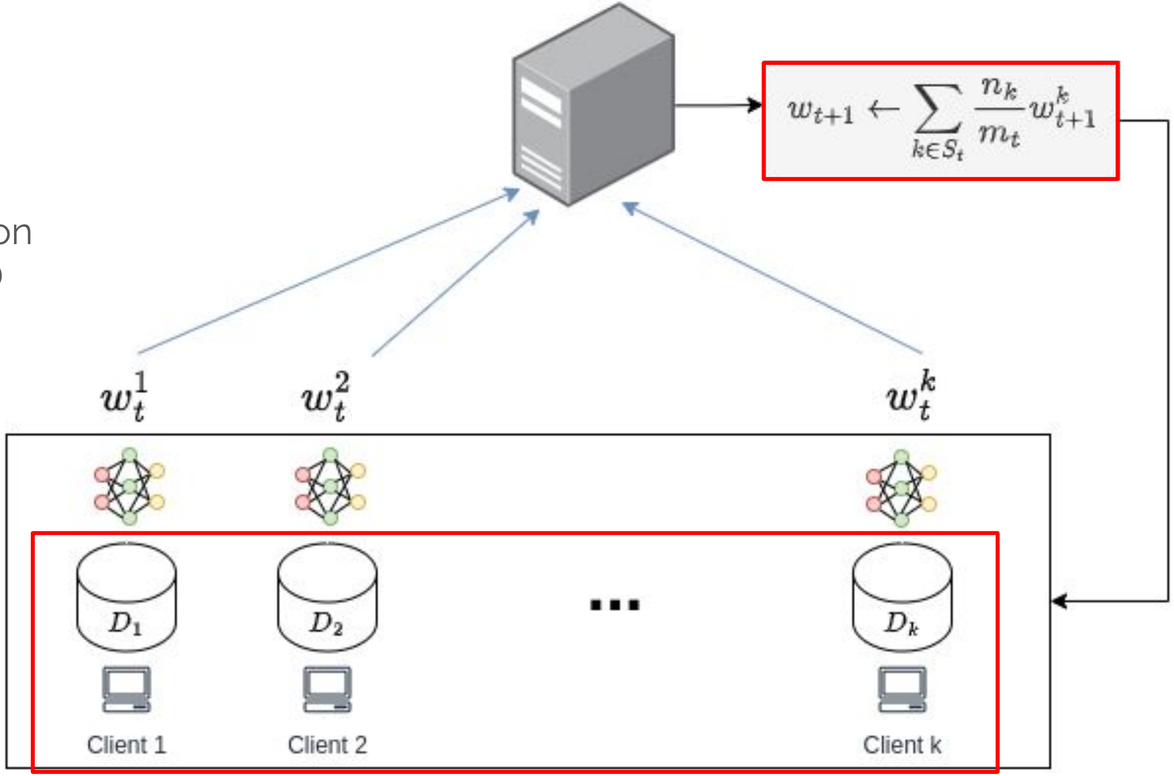
Centralized Machine Learning



Federated Learning

FedAvg [1]:

- **Random** client selection
- Local iterations of **SGD**
- **Server aggregation**



Outline

1. ML vs FML
2. Research questions
3. Zero-Knowledge proofs
4. Proposed framework
5. Experiments and results
6. Possible improvements

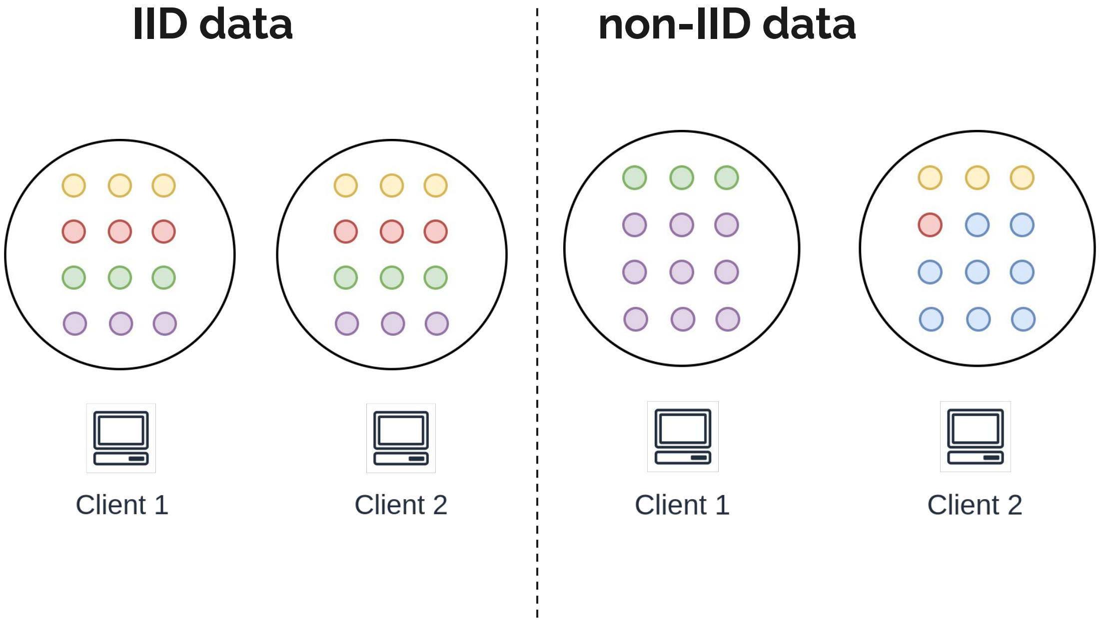
Research Questions

- How to **quantify** a client contribution?
- How is training affected by **malicious** contributions?
- How can the server verify that a client's declared contribution is **trustworthy**?

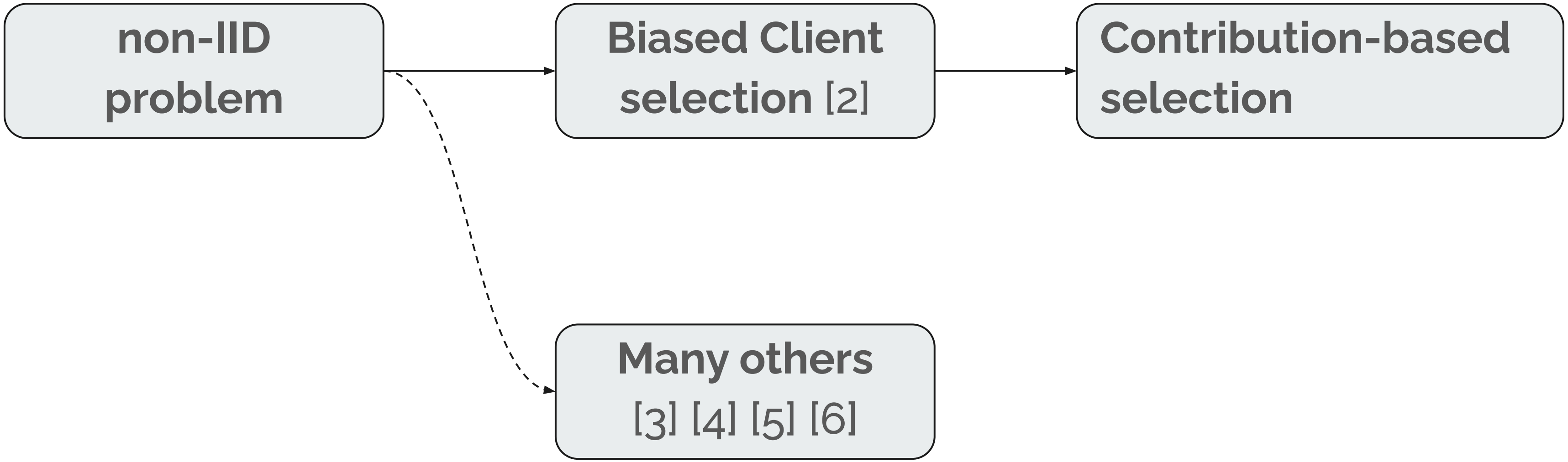
?

The IID problem

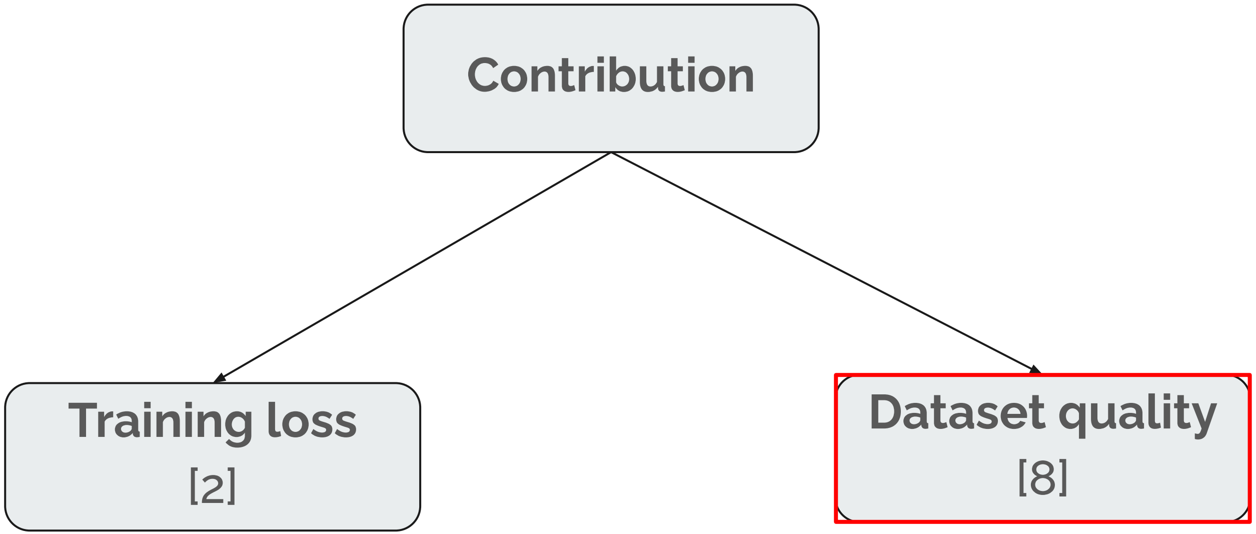
- High **client drift**: local models diverge from global objective
- **Slower** convergence and potential model **bias**.



Solutions to non-IID data



How to quantify a client contribution?

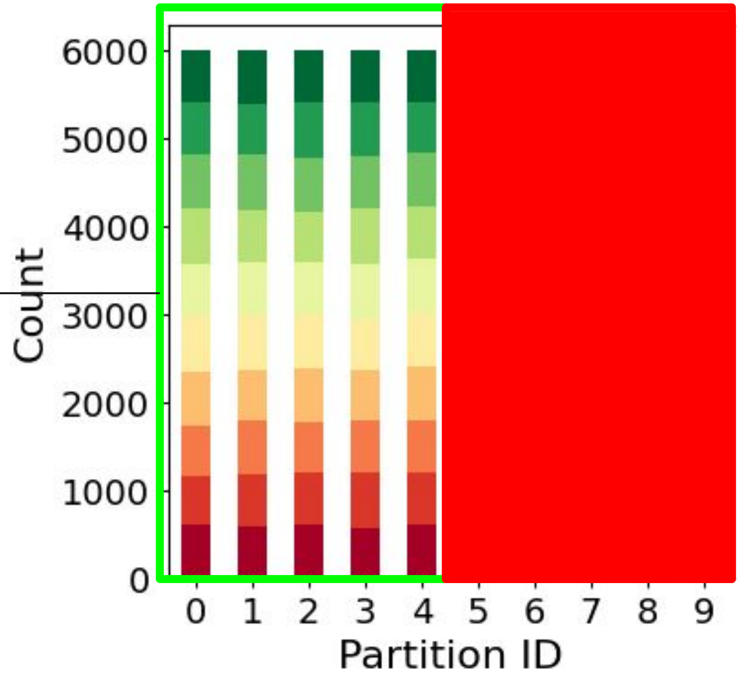


Dataset score

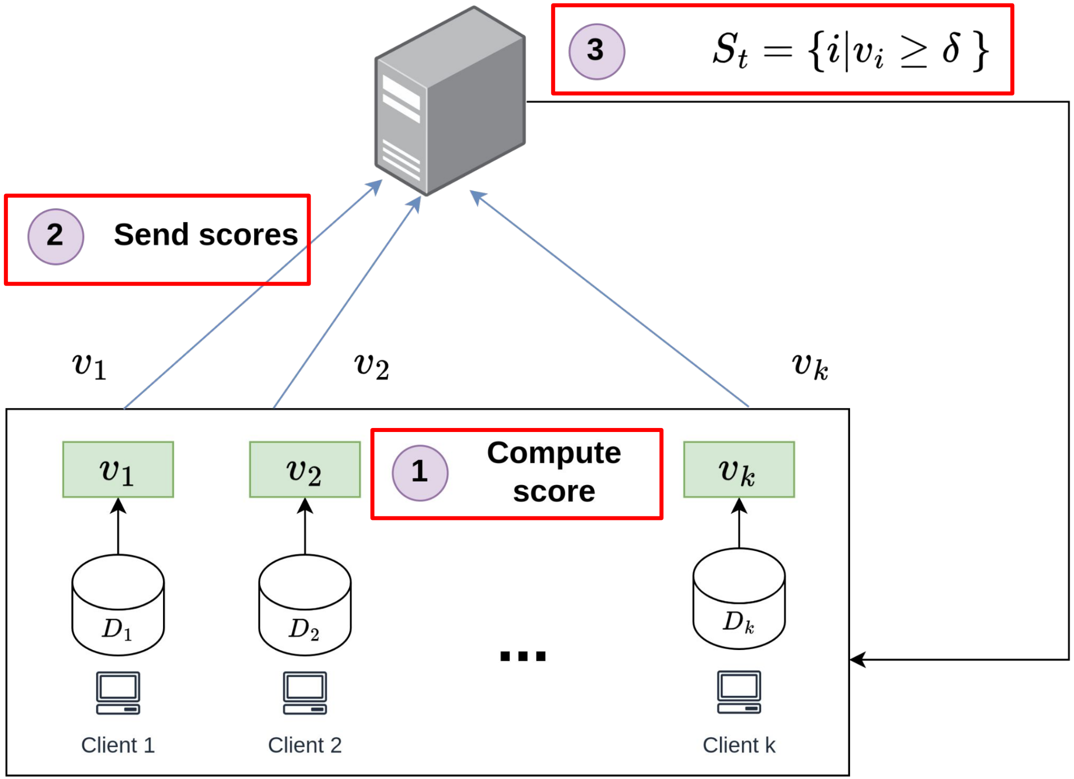
$$v = \phi(D)$$

- Quantify the **IIDness**
- Clients with **“similar”** dataset have **near identical** score.

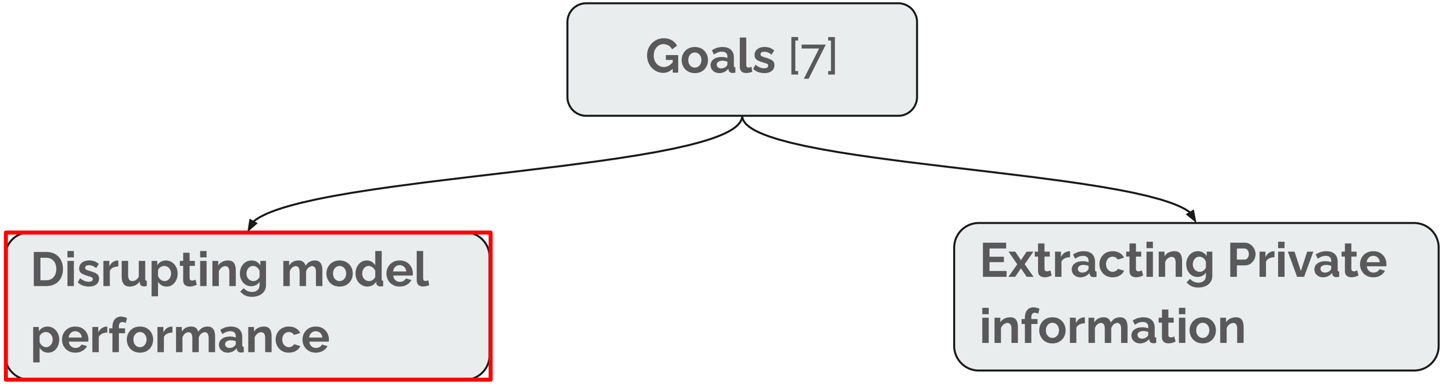
This set of clients is “better”



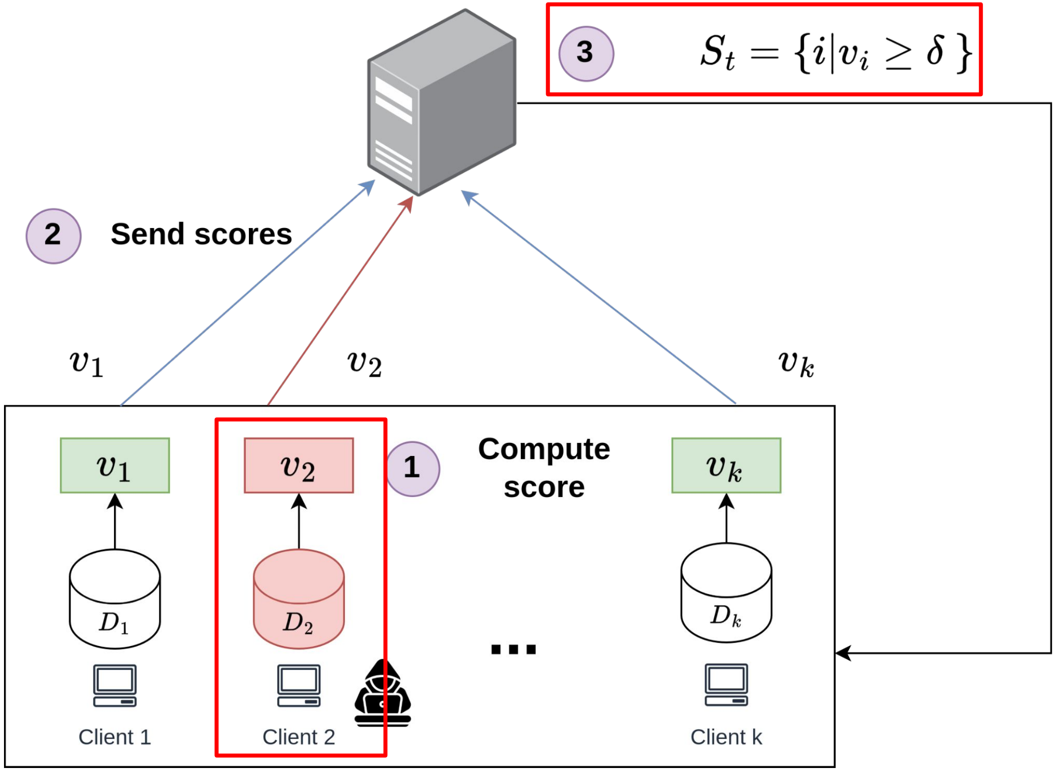
Score-based selection



Malicious actors



What if?



3
$S_t = \{i | v_i \geq \delta\}$
2
Send scores
v_1
v_2
v_k
1
Compute score
v_1
v_2
v_k
D_1
D_2
D_k
...
Client 1
Client 2
Client k

Trustworthy contribution

Can a client provide **evidence** of its **contribution** without **revealing private information**?

Zero-Knowledge proofs

Outline

1. ML vs FML
2. Research questions
3. Zero-Knowledge proofs
4. Proposed framework
5. Experiments and results
6. Possible improvements

Zero-Knowledge Proofs

*"A cryptographic tool allowing a **Prover** to convince a **Verifier** about the validity of a statement without revealing any sensitive information".*

- **Completeness:** if the prover is telling the truth, it will eventually convince the verifier.
- **Soundness:** if the prover is **not** telling the truth, the verifier rejects the proof.
- **Zero-Knowledge:** the verifier learns **nothing** beyond the statement's validity.

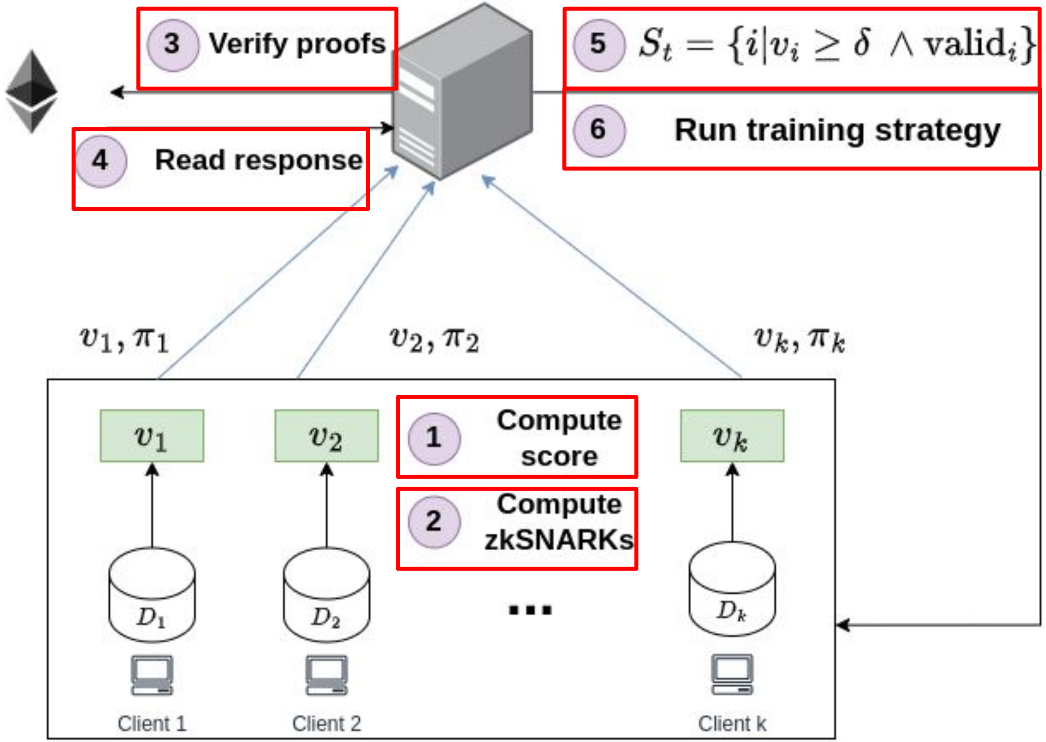
zkSNARKs

- A tool to efficiently generate ZKPs for arbitrary functions.
- Mainly used in **Ethereum Blockchain**.
- **Properties:**
  - **zk**: hides input
  - **S**uccint: short proofs, quickly verifiable.
  - **N**on-interactive: just the proof is exchanged
  - **AR**gument-of-knowledge: proves you know the input

Outline

1. ML vs FML
2. Research questions
3. Zero-Knowledge proofs
4. Proposed framework
5. Experiments and results
6. Possible improvements

Proof of score



Enhancing training speed

We want an algorithm able to:

- **bias** client selection
- provide guarantee of **contribution validity**
- enhancing the **training convergence**

Power of Choice + **Score based selection** + **ZK** = **PoCZk**

Outline

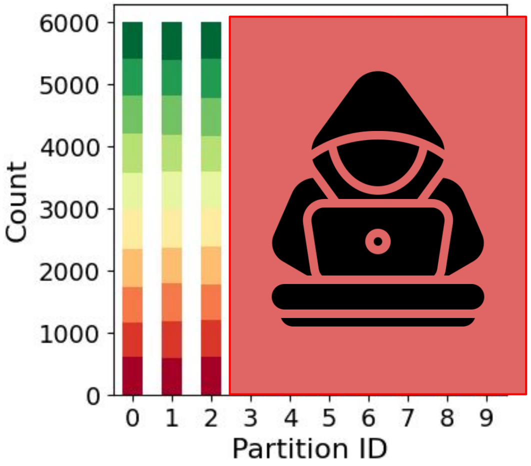
1. ML vs FML
2. Research questions
3. Zero-Knowledge proofs
4. Proposed framework
5. Experiments and results
6. Possible improvements

Experiment

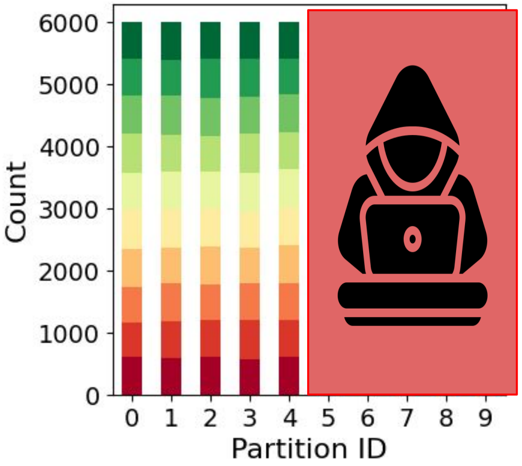
- **Testbed**: Fedora 41, 8 GB RAM, I5-8250U 3.4 GHz
- **Framework**: Flower for FL and ZoKrates for zkSNARK
- **Dataset**: MNIST, FMNIST, CIFAR10
- **Metrics**: Centralized accuracy and communication rounds.
- **Baselines**: $\underbrace{\textit{FedAvg, FedAvgM, FedAdam, FedProx,}}_{\textbf{Random}\ \text{selection}}$ $\underbrace{\textit{ContAvg, ZkAvg}}_{\textbf{Biased}\ \text{selection}}$

Data partitioning

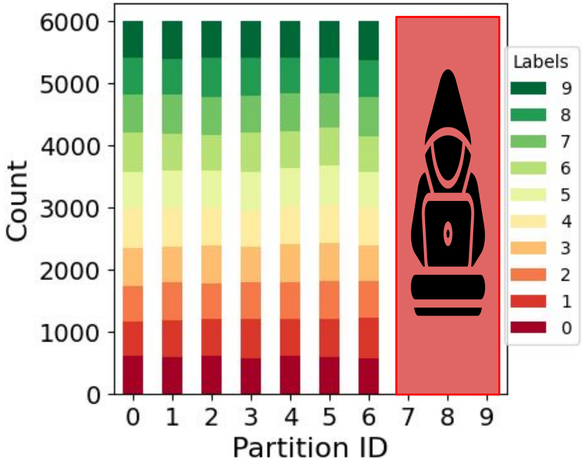
iid = 30 %



iid = 50 %

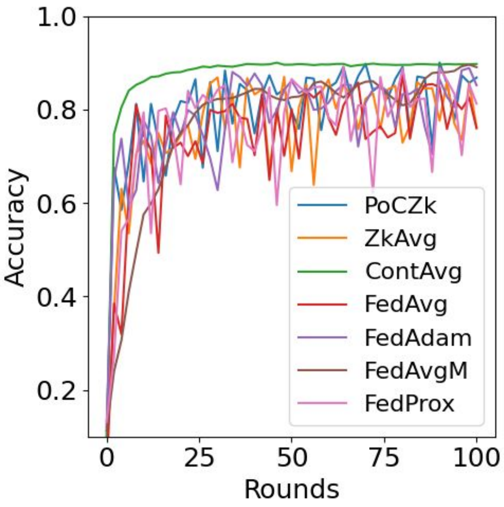


iid = 70 %

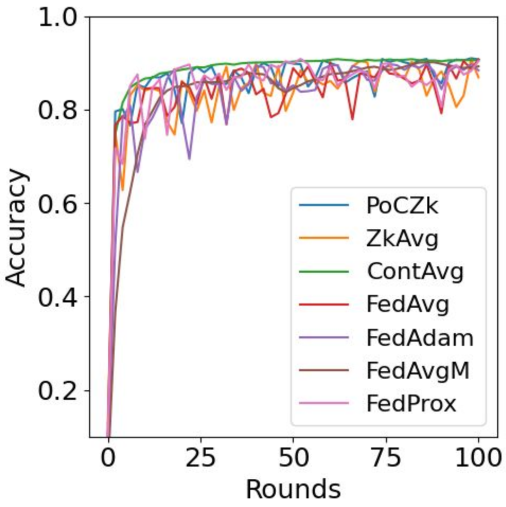


Training rounds (honest)

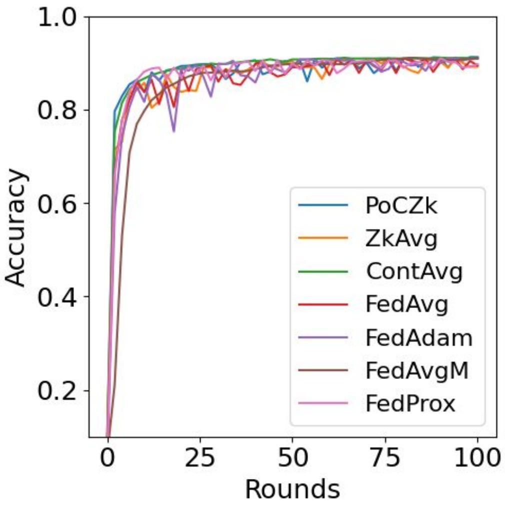
iid = 30 %



iid = 50 %

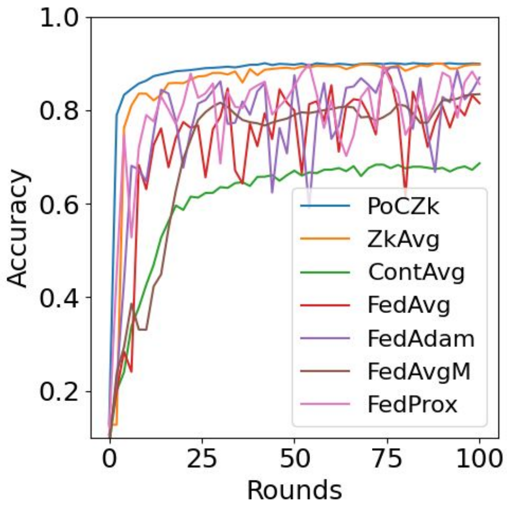


iid = 70 %

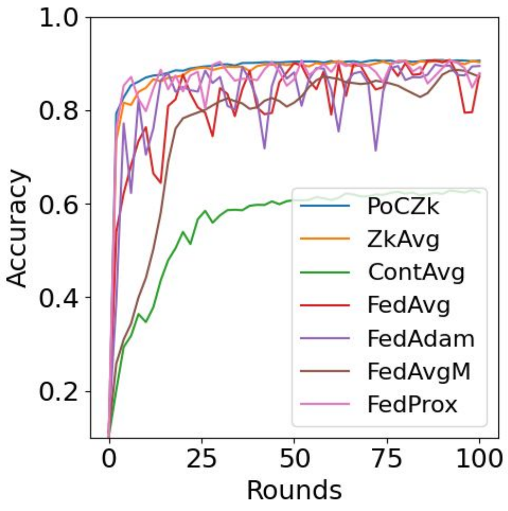


Training rounds (dishonest)

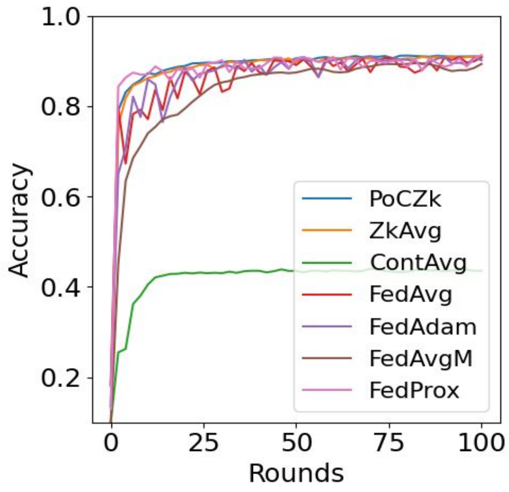
iid = 30 %



iid = 50 %



iid = 70 %



Convergence speed

Algorithm	MNIST (98%) 0.3	MNIST (98%) 0.5	MNIST (98%) 0.7	FMNIST (89%) 0.3	FMNIST (89%) 0.5	FMNIST (89%) 0.7	CIFAR10 (45%) 0.3	CIFAR10 (45%) 0.5	CIFAR10 (45%) 0.7
FedAvg	28	16	12	60	20	16	∞	36	28
FedAvgM	82	44	36	∞	56	30	74	42	56
FedAdam	28	14	10	30	14	10	∞	70	48
FedProx	22	14	**8**	22	**4**	**4**	34	28	14
PoCZk	**6**	**8**	**8**	**8**	6	8	**16**	**12**	**8**
ZkAvg	18	10	**8**	16	12	**8**	34	16	14
ContAvg	∞	∞	∞	∞	∞	∞	∞	∞	∞

Outline

1. ML vs FML
2. Research questions
3. Zero-Knowledge proofs
4. Proposed framework
5. Experiments and results
6. Possible improvements

Possible improvements

- **Enhanced ZKP**: Develop Zero-Knowledge Proofs (ZKP) to support floating-point and complex computations for richer client metrics
- **Scalability testing**: Experiment with larger client numbers and real-world scenarios (e.g., IoT devices) to assess practical performance.
- **Reward-based strategy**: Develop reward strategies through evidence-based incentives by means of Smart Contract.

References

1. McMahan, Brendan, et al. "*Communication-efficient learning of deep networks from decentralized data.*" Artificial intelligence and statistics. PMLR, 2017.
2. Cho, Yae Jee, Jianyu Wang, and Gauri Joshi. "*Client selection in federated learning: Convergence analysis and power-of-choice selection strategies.*" arXiv preprint arXiv:2010.01243 (2020).
3. Zhu, Hangyu, et al. "*Federated learning on non-IID data: A survey.*" Neurocomputing 465 (2021): 371-390.
4. Reddi, Sashank, et al. "*Adaptive federated optimization.*" arXiv preprint arXiv:2003.00295 (2020).
5. Zhao, Yue, et al. "*Federated learning with non-iid data.*" arXiv preprint arXiv:1806.00582 (2018).
6. Itahara, Sohei, et al. "*Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data.*" IEEE Transactions on Mobile Computing 22.1 (2021): 191-205.
7. Mothukuri, Viraaji, et al. "*A survey on security and privacy of federated learning.*" Future Generation Computer Systems 115 (2021): 619-640.
8. Ye, Rui, et al. "*Feddisco: Federated learning with discrepancy-aware collaboration.*" International Conference on Machine Learning. PMLR, 2023.

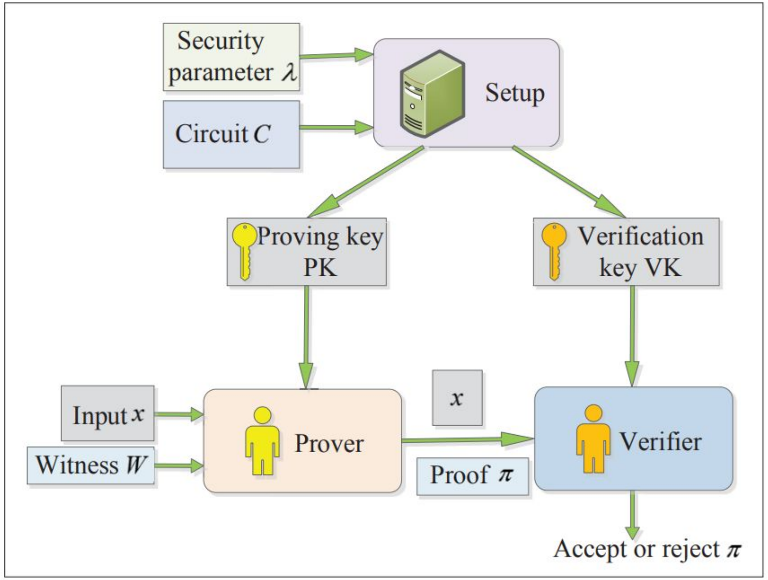
Thank you for the attention
:)

Appendix

1. Zero-Knowledge
2. Dataset score
3. Merkle Tree commitment
4. Power Of Choice with ZK
5. Additional results

zkSNARKs: need of a trusted setup

Turn a problem into a function → Turn the function into R1CS → Generate a proof of satisfiability of the R1CS



Appendix

1. Zero-Knowledge
2. Dataset score
3. Merkle Tree commitment
4. Power Of Choice with ZK
5. Additional results

Dataset score

$$v = \phi(D) = \beta_1 \underbrace{\sum_{i=1}^{k} (c_i - \mu_L)^2}_{\text{Variance}} + \beta_2 \overbrace{\sum_{i=1}^{k} \mathbb{I}(c_i, \mathrm{T})}^{\text{Class Diversity}}$$

$$\mathbb{I}(c_i, \mathrm{T}) = \begin{cases} 1 & \text{if } c_i \geq \mathrm{T} \\ 0 & \text{otherwise} \end{cases}$$

Appendix

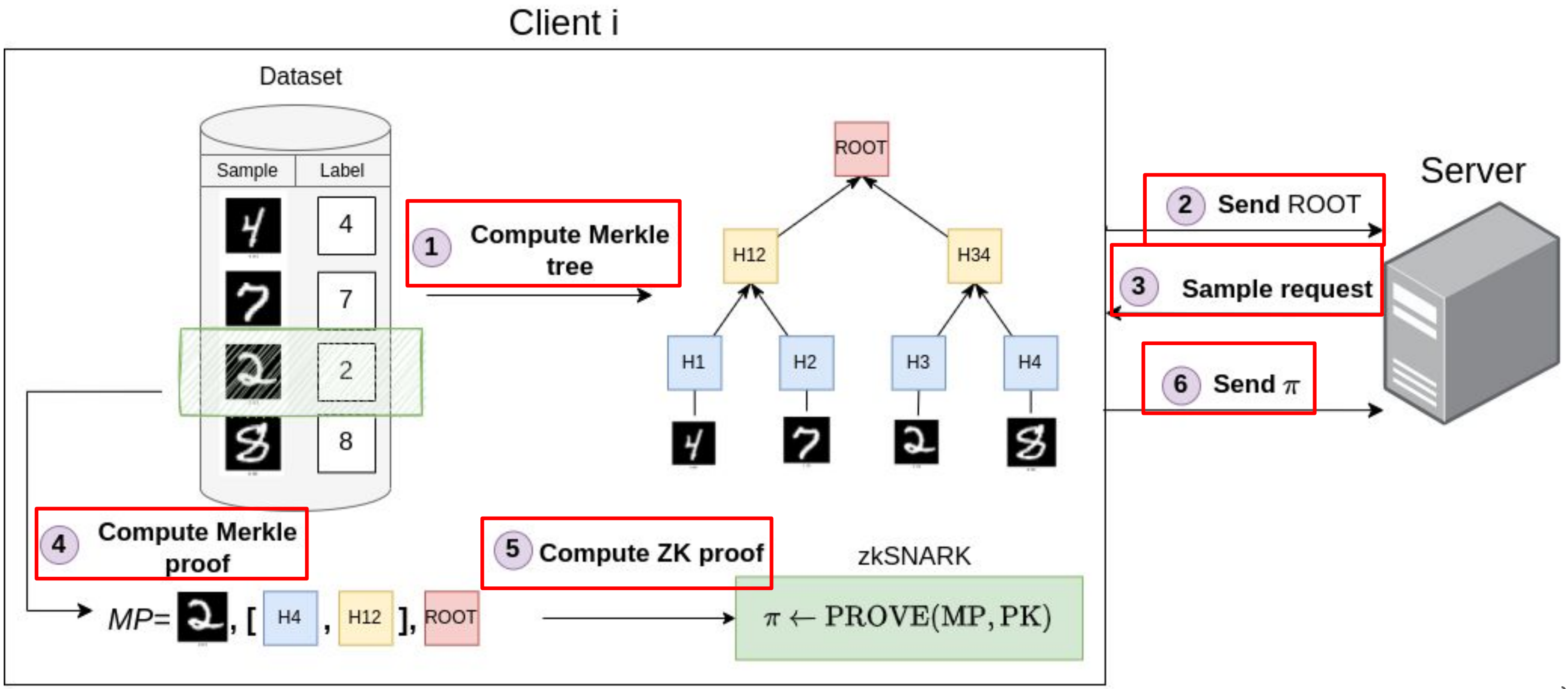
1. Zero-Knowledge
2. Dataset score
3. Merkle Tree commitment
4. Power Of Choice with ZK
5. Additional results

Ensuring Data authenticity

How to verify datasets **authenticity efficiently**?

Merkle tree + ZK

Merkle proof of dataset with ZK

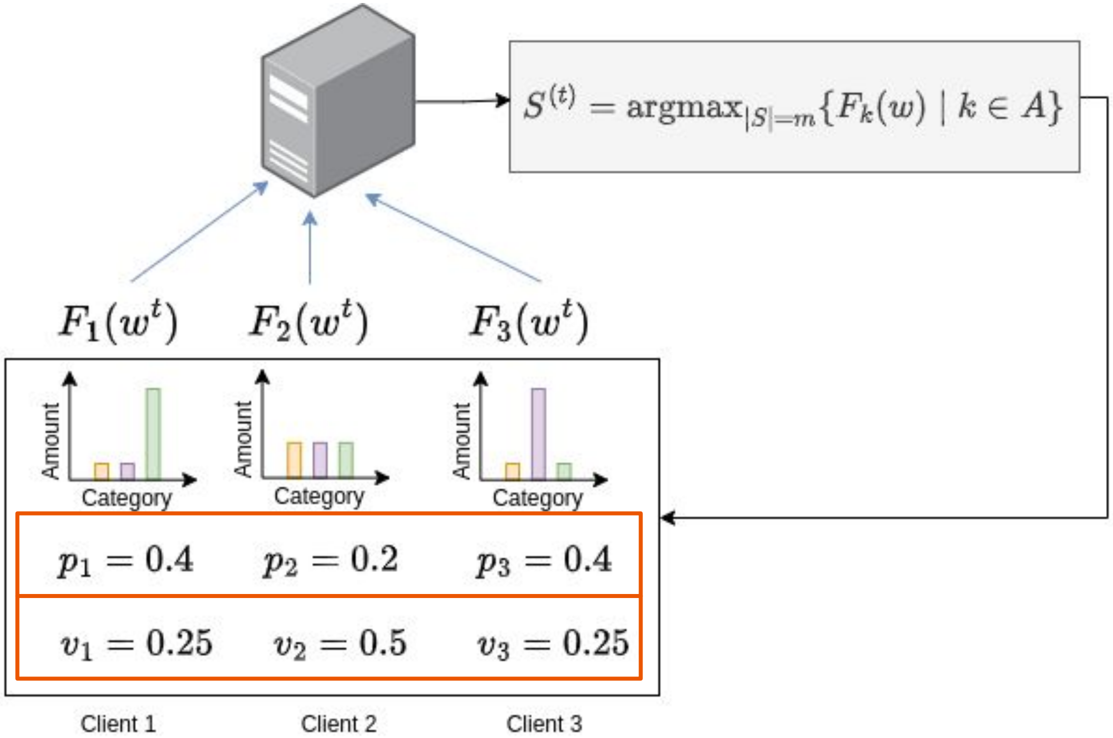


Appendix

1. Zero-Knowledge
2. Dataset score
3. Merkle Tree commitment
4. Power Of Choice with ZK
5. Additional results

Power of Choice with ZK (PoCZk)

1. **Sample clients**: Select *d* clients based on p_k.
2. **Estimate losses**: Clients compute and send local losses $F_k(w)$.
3. **Select Top m**: Choose *m* client with highest $F_k(w)$ for training.



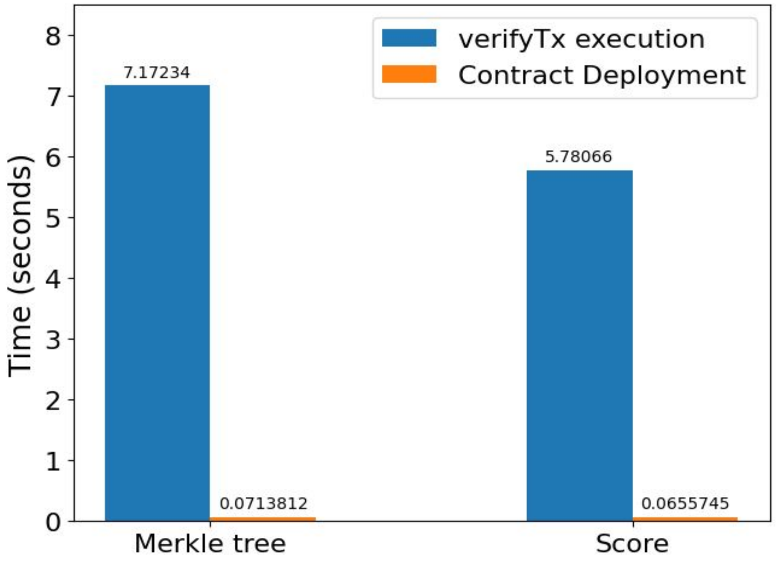
Appendix

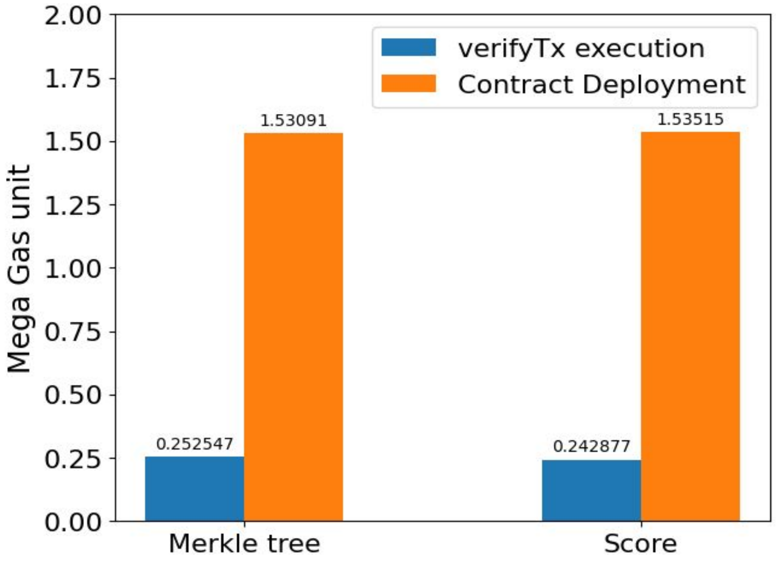
1. Zero-Knowledge
2. Dataset score
3. Merkle Tree commitment
4. Power Of Choice with ZK
5. Additional results

Accuracy gain

Algorithm	Accuracy	
	Honest	Dishonest
PoCZk	**90.04%**	**90.02%**
ZkAvg	86.98% (↓ **3.06%**)	89.94% (↓ **0.08%**)
ContAvg	90.00% (↓ **0.04%**)	68.62% (↓ **21.40%**)
FedAvg	87.77% (↓ **2.27%**)	89.45% (↓ **0.57%**)
FedAdam	88.84% (↓ **1.20%**)	89.11% (↓ **0.91%**)
FedAvgM	89.57% (↓ **0.47%**)	83.39% (↓ **6.63%**)
FedProx	89.12% (↓ **0.92%**)	89.78% (↓ **0.24%**)

Smart Contract cost





Analysis

- **Low iid_ratio + dishonest nodes**: *PoCZk* and *ZkAvg* outperform others; *PoCZk* reduces rounds by **7.5x** vs *FedAvg*.
- **Honest nodes**: *ContAvg* is the best, reducing rounds by **8.5x** vs. *FedAvg*.
- *Random-based selection algorithms* show moderate performance but suffer from instability because of poor client selection.