45th International Conference on Software Engineering
ICSE2023
Melbourne Convention and Exhibition Centre
14-20 May 2023

FedSlice: Protecting Federated Learning Models from Malicious Participants with Model Slicing

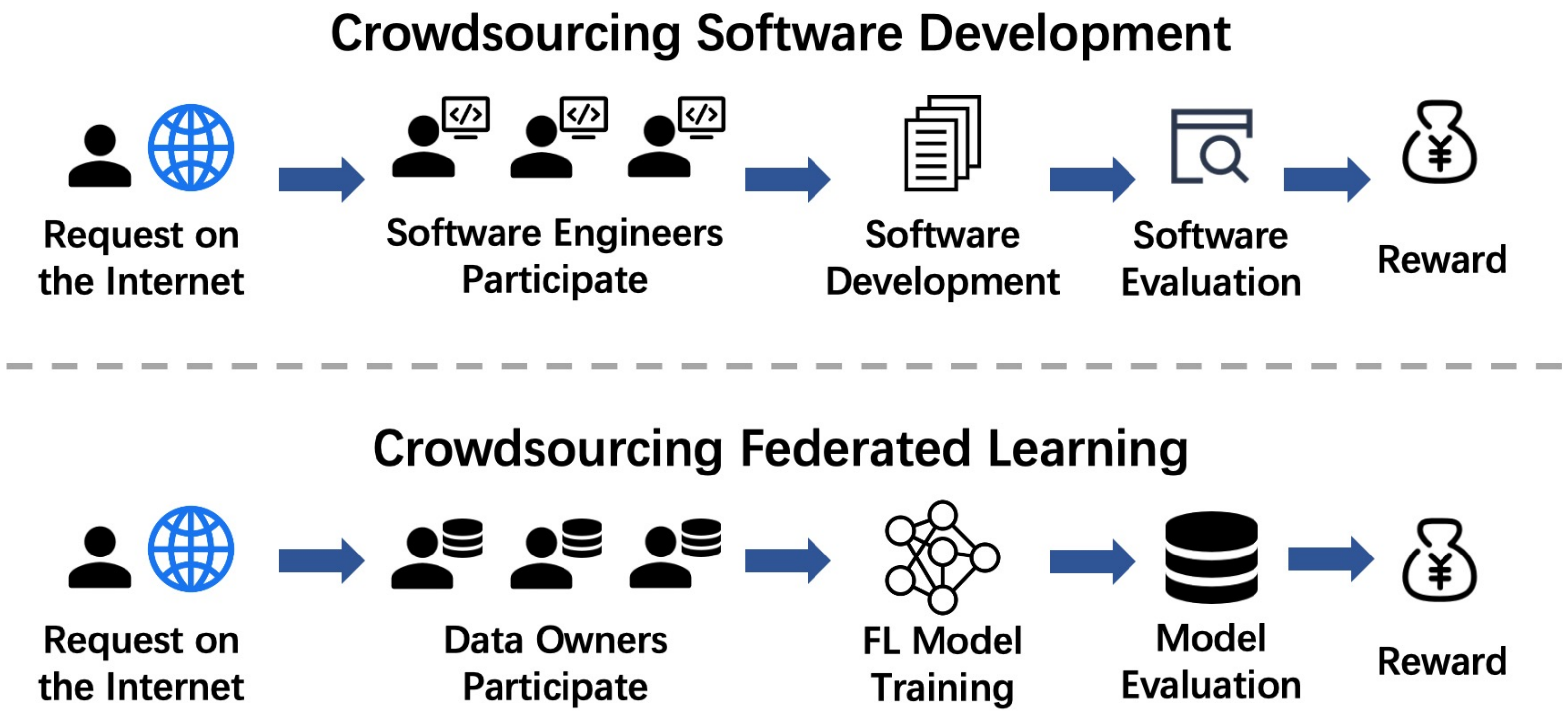
Ziqi Zhang, Yuanchun Li, Bingyan Liu, Yifeng Cai,
Ding Li, Yao Guo, Xiangqun Chen

Software 2.0: DNN v.s. Traditional Program

	DNN	Programs
Similarity	Set of data and instructions organized by a pre-defined order, wrote by a specific program language, execute behavior logics defined by developers	
Software Development	Developers collect data and train the model	Developers write code instructions
Software Behavior	Defined by model weights	Defined by program instructions
Software Understand	Hard to understand the mechanism of single parameter	Easy to understand functionalities of instructions by inverse engineering
Software Debug	Update data and retrain models	Change buggy instructions

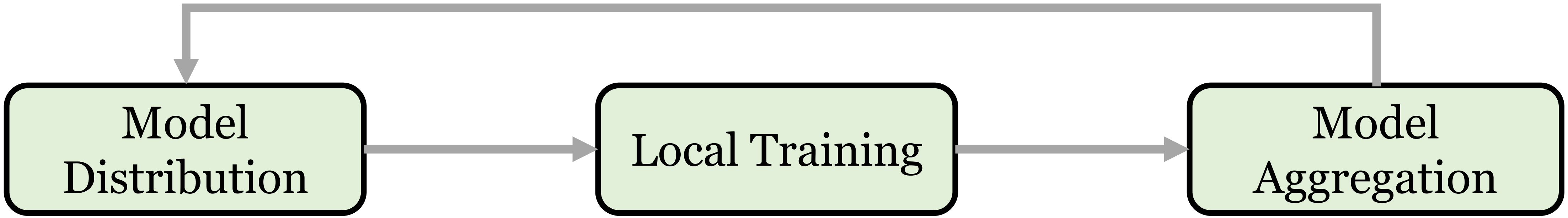
Crowdsourcing Federated Learning (CFL)

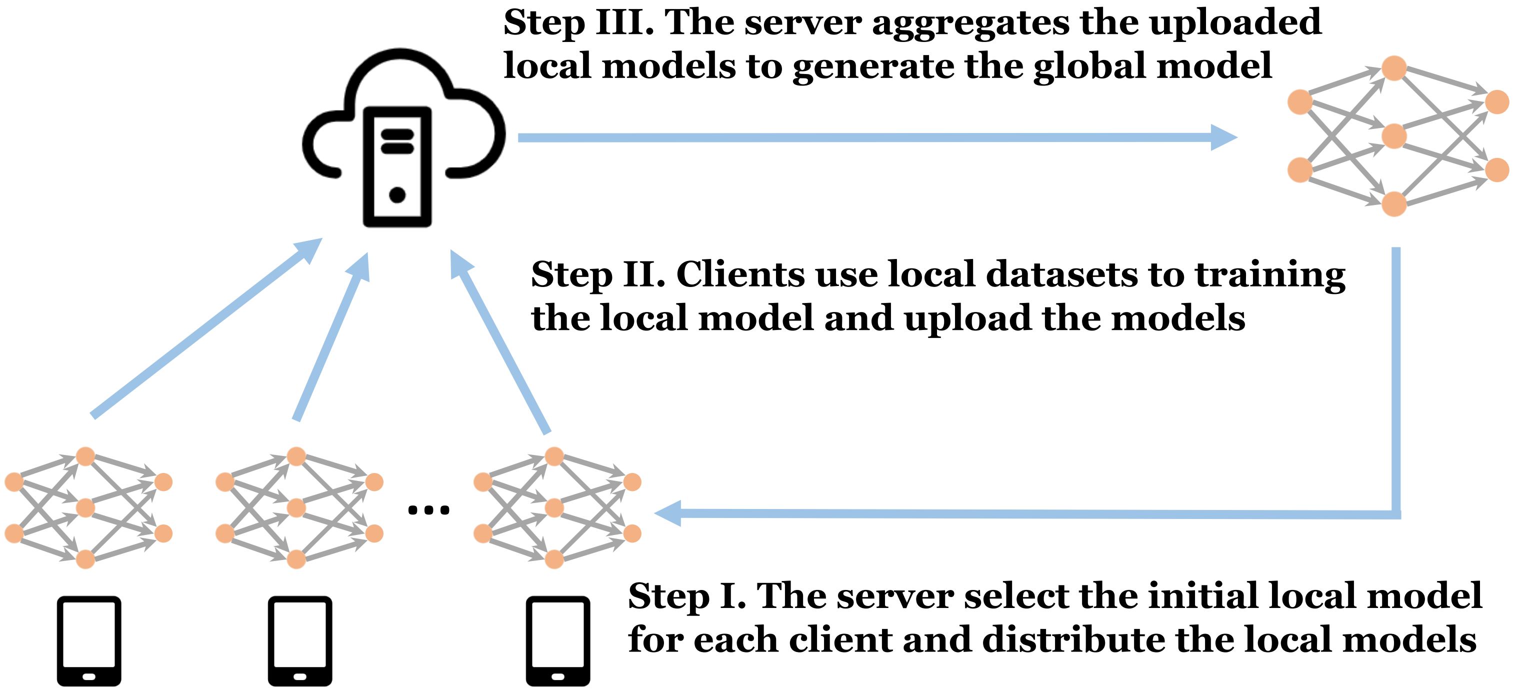
Crowdsourcing federated learning is a form of crowdsourcing development scheme for DNN software



Feng et al. CrowdFL: A Marketplace for Crowdsourced Federated Learning. AAAI 2022
Pandey et al. A crowdsourcing framework for on-device federated learning. IEEE Transactions on Wireless Communications 2020
Tong et al. Federated learning in the lens of crowdsourcing. IEEE Data Eng. Bull. 2020.

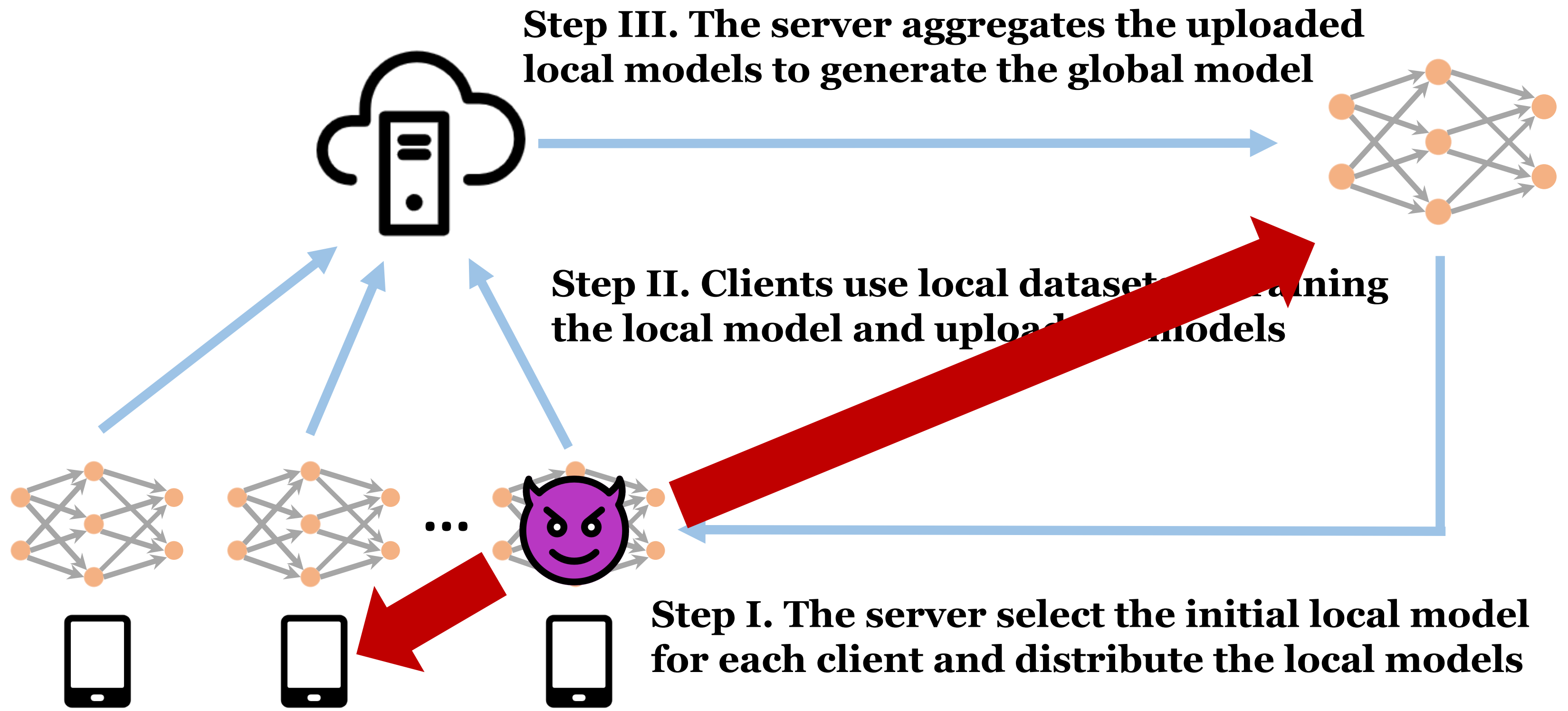
Typical CFL Pipeline





Security Threats from Malicious Participants

Adversary participants may attack the trained server model and the data of other participants



Security Threats from Malicious Participants

Adversary participants may attack the trained server model and the data of other participants

Free-Rider Attack

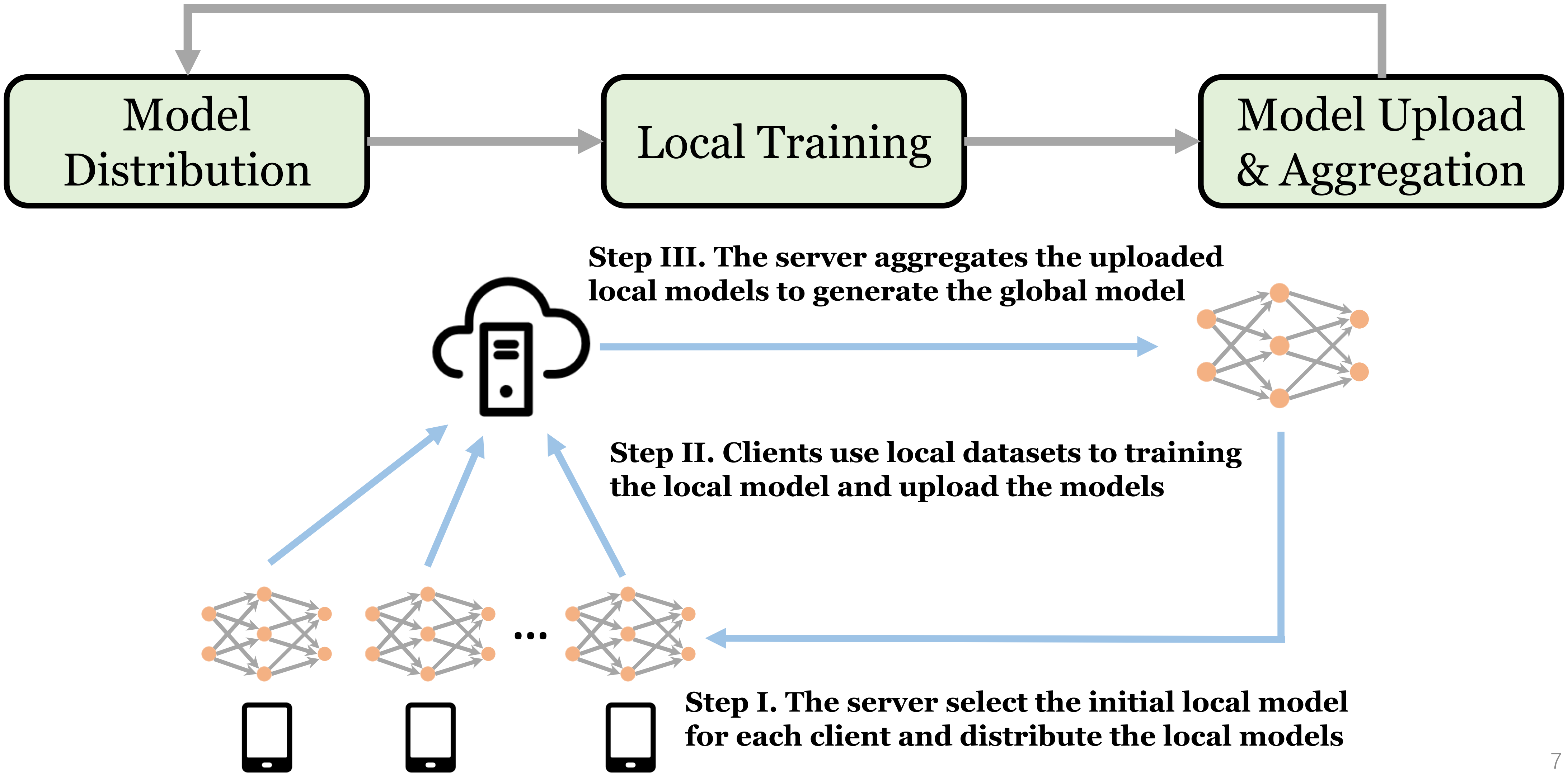
Adversarial Attack

Membership Inference

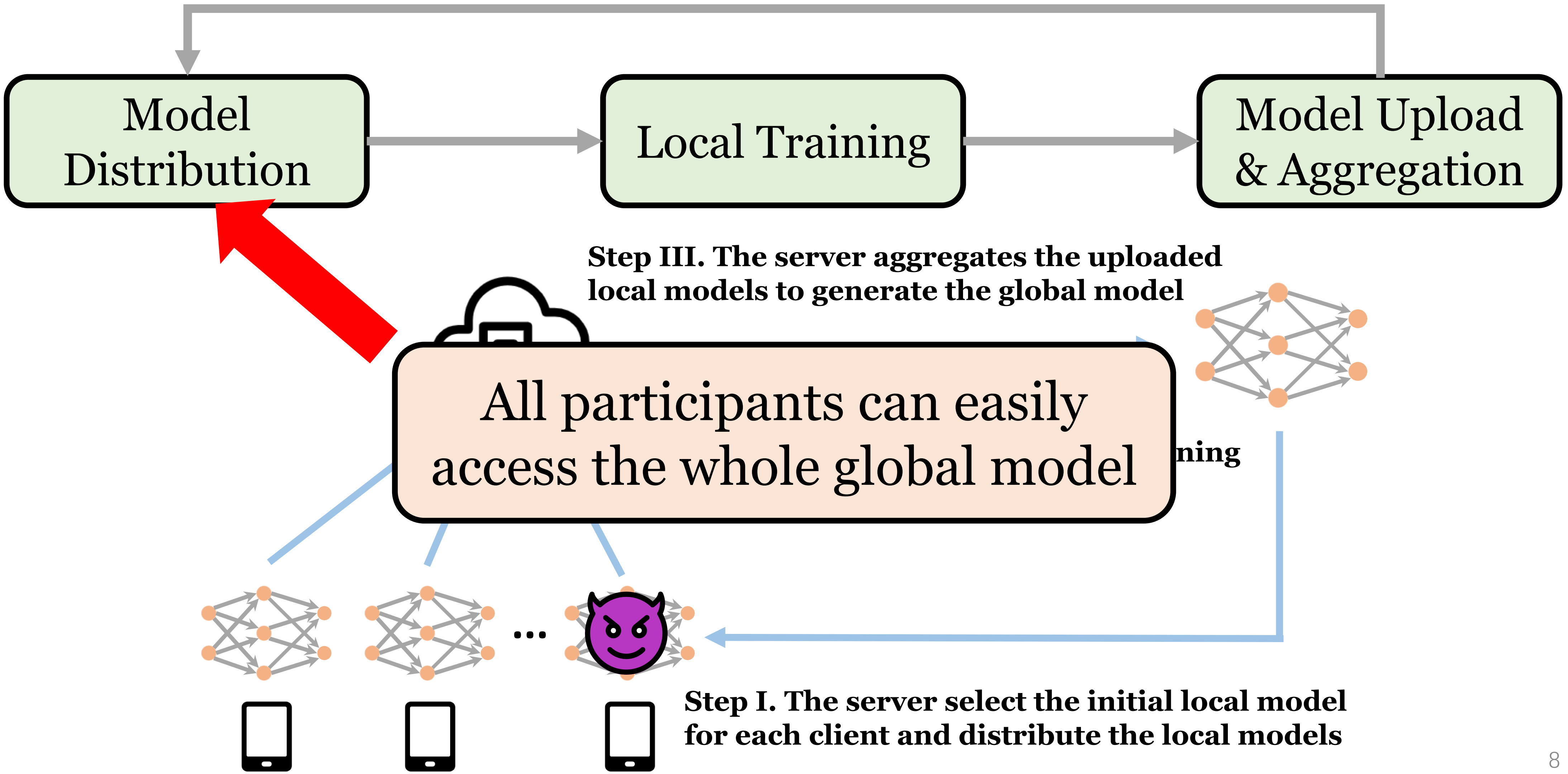
Deep Gradient Leakage

Fraboni et al. Free-rider attacks on model aggregation in federated learning. AISTATS 2021
Madry et al. Towards deep learning models resistant to adversarial attacks. ICLR 2018
Shokri et al. Membership Inference Attacks against Machine Learning Models. S&P 2017
Zhu et al. Deep leakage from gradients. NeurIPS 2019

Fundamental Cause of Attacks



Fundamental Cause of Attacks



Limitation of Existing Defenses

Solution 1
Encrypt the model weights

Slow the computation speed by 1,000×

Solution 2
Shield the model with Trusted Execution Environments

Slow the computation speed by 36×

Fundamental Reason

Existing defenses try to protect all the model weights (the whole large model) and introduce high overhead

Tramer et al. Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware. *ICLR 2019*
Gilad-Bachrach et al. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. ICML 2016

Goal and Insight

Goal

Defend against dishonest participants and efficiently protect server model against four attacks without using encryption or TEEs

Insight

Partition the model into different parts and each participant only access what is allowed.

Participants may not need the whole model to contribute training

Technical Challenges

How to modify the model while preserving the CFL pipeline?

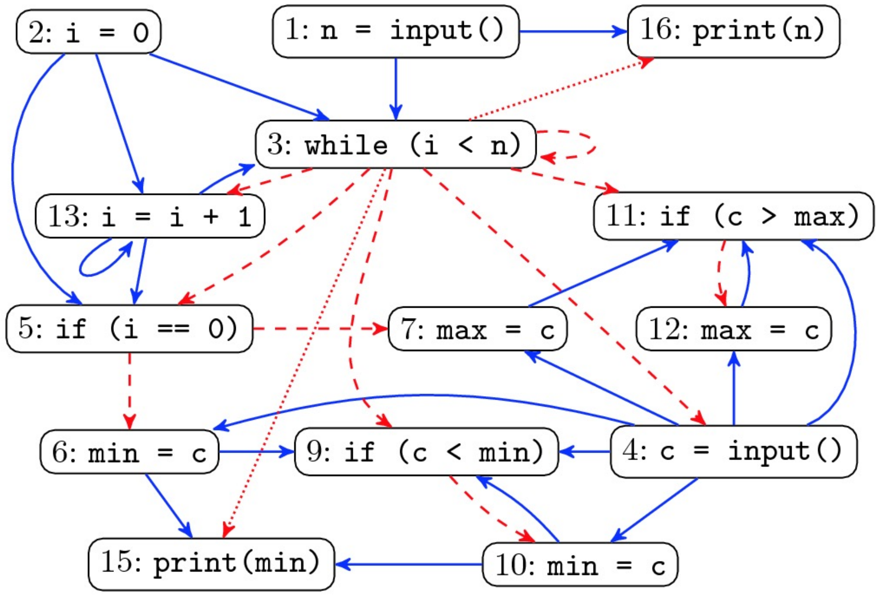
C1: How to separate the server model into different parts?

C2: How to recompose the model fragments from participants?

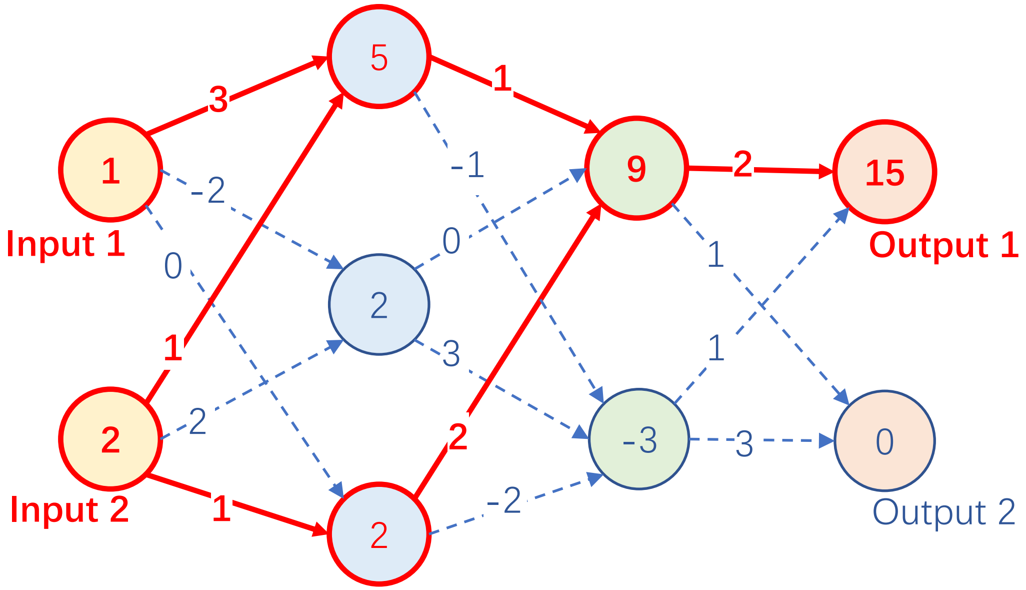
Motivation: Model Slicing

Program Slice

```
1   n = input();
2   i = 0;
3   while (i < n) {
4     c = input();
5     if (i == 0) {
6       min = c;
7       max = c;
8     }
9     if (c < min)
10      min = c;
11    if (c > max)
12      max = c;
13    i = i + 1;
14  }
15  print(min);
16  print(n);
```



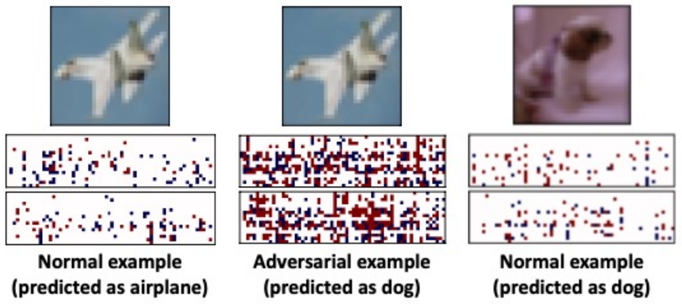
Model Slice



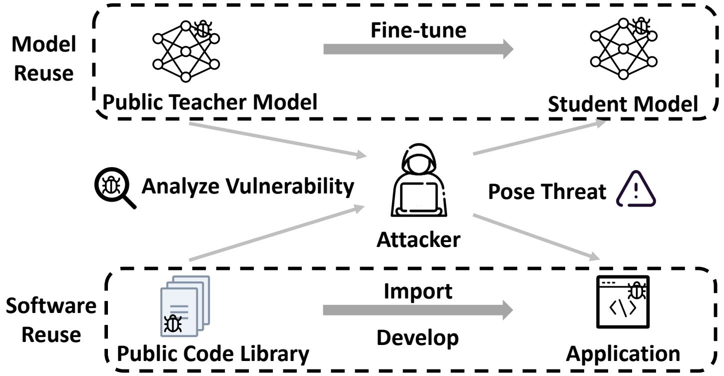
Mark Weiser. Program slicing. TSE 1984
Zhang et al. Dynamic slicing for deep neural networks. ESEC/FSE 2020

Model Slicing for Security

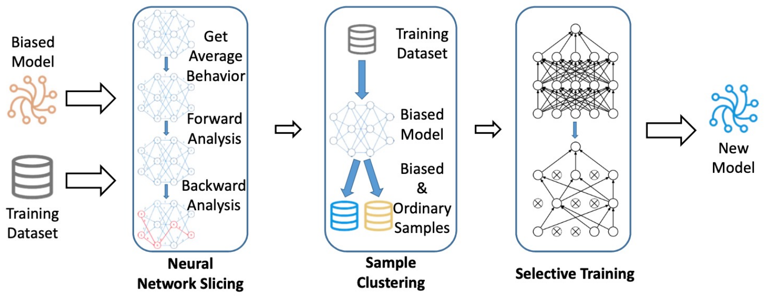
As program slicing can help to improve the security of traditional programs, model slicing can improve the security of DNN models



Abnormal model behavior detection in DNN inference



Secure model reuse in transfer learning

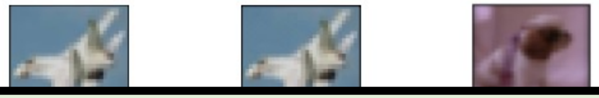


Correct fairness faults of DNN models

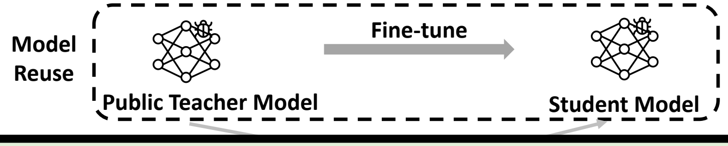
Zhang et al. Dynamic slicing for deep neural networks. ESEC/FSE 2020
Zhang et al. ReMoS: Reducing Defect Inheritance in Transfer Learning via Relevant Model Slicing. ICSE 2022
Gao et al. FairNeuron: Improving Deep Neural Network Fairness with Adversary Games on Selective Neurons. ICSE 2022

Model Slicing for Security

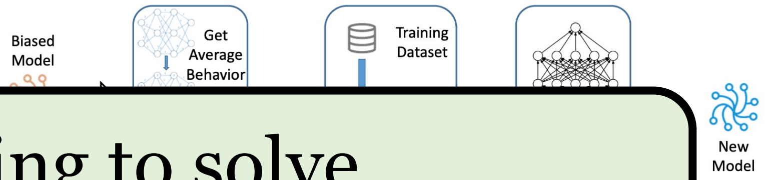
As program slicing can help to improve the security of traditional programs, model slicing can improve the security of DNN models



Abnormal model behavior detection in DNN inference



Secure model reuse in transfer learning



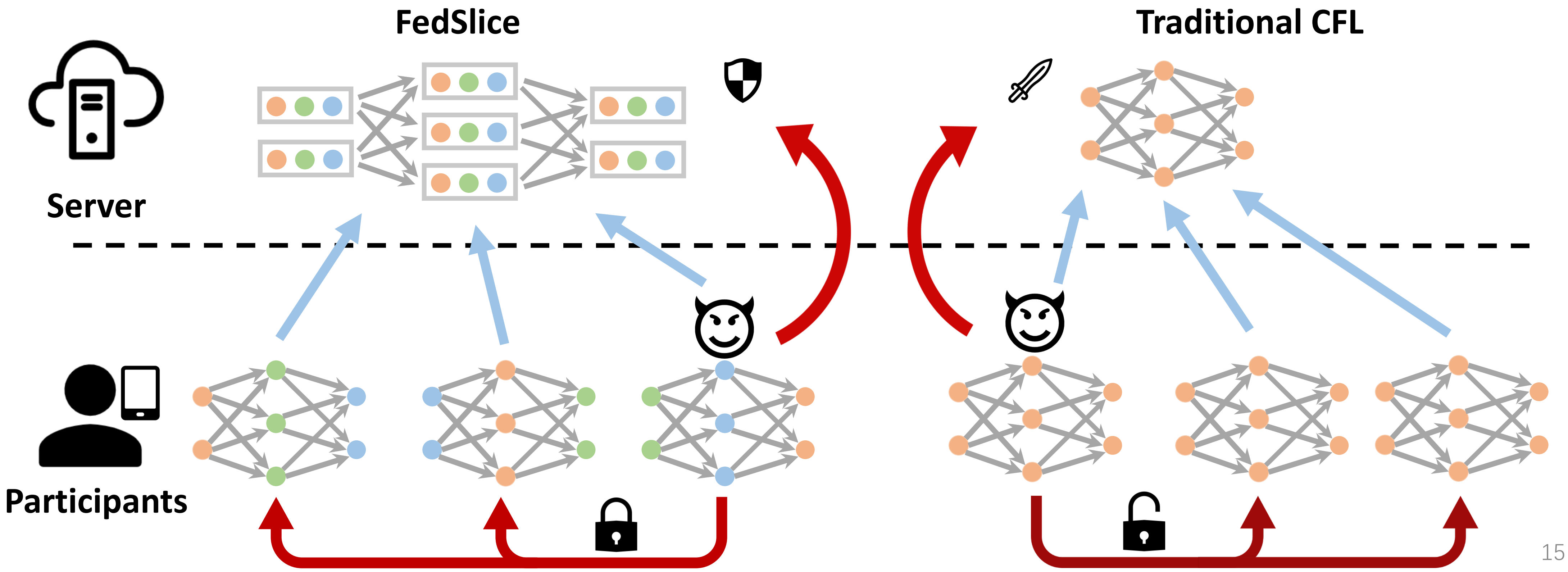
Correct fairness faults of DNN models

In this work, we want to use model slicing to solve the security issues of CFL

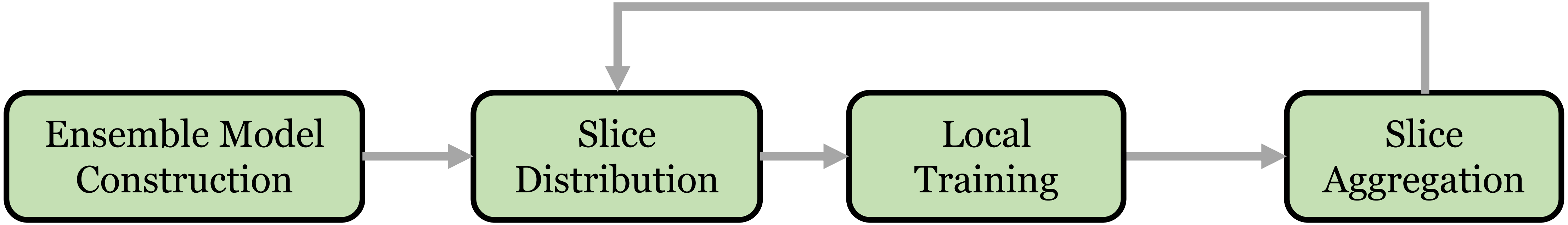
Zhang et al. Dynamic slicing for deep neural networks. ESEC/FSE 2020
Zhang et al. ReMoS: Reducing Defect Inheritance in Transfer Learning via Relevant Model Slicing. ICSE 2022
Gao et al. FairNeuron: Improving Deep Neural Network Fairness with Adversary Games on Selective Neurons. ICSE 2022

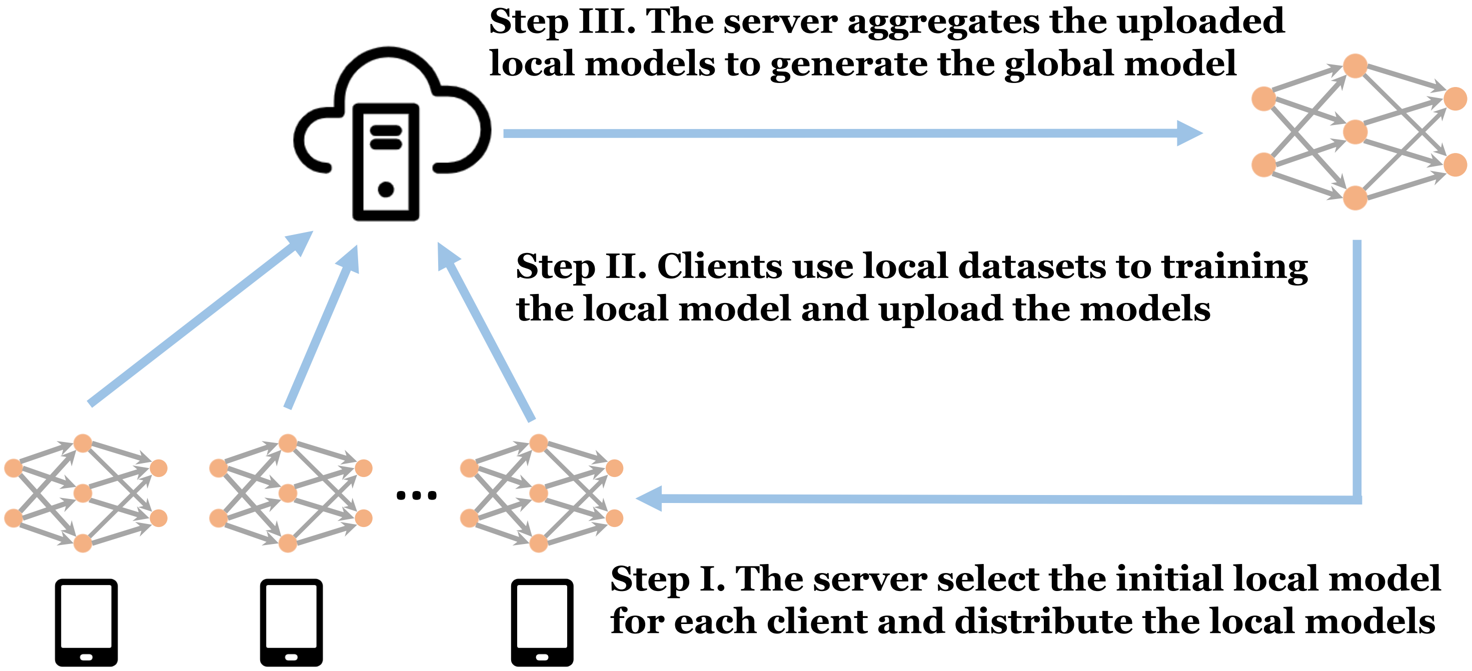
Our Solution: FedSlice

Partition the server's model into multiple slices and distribute different model slices to different participants

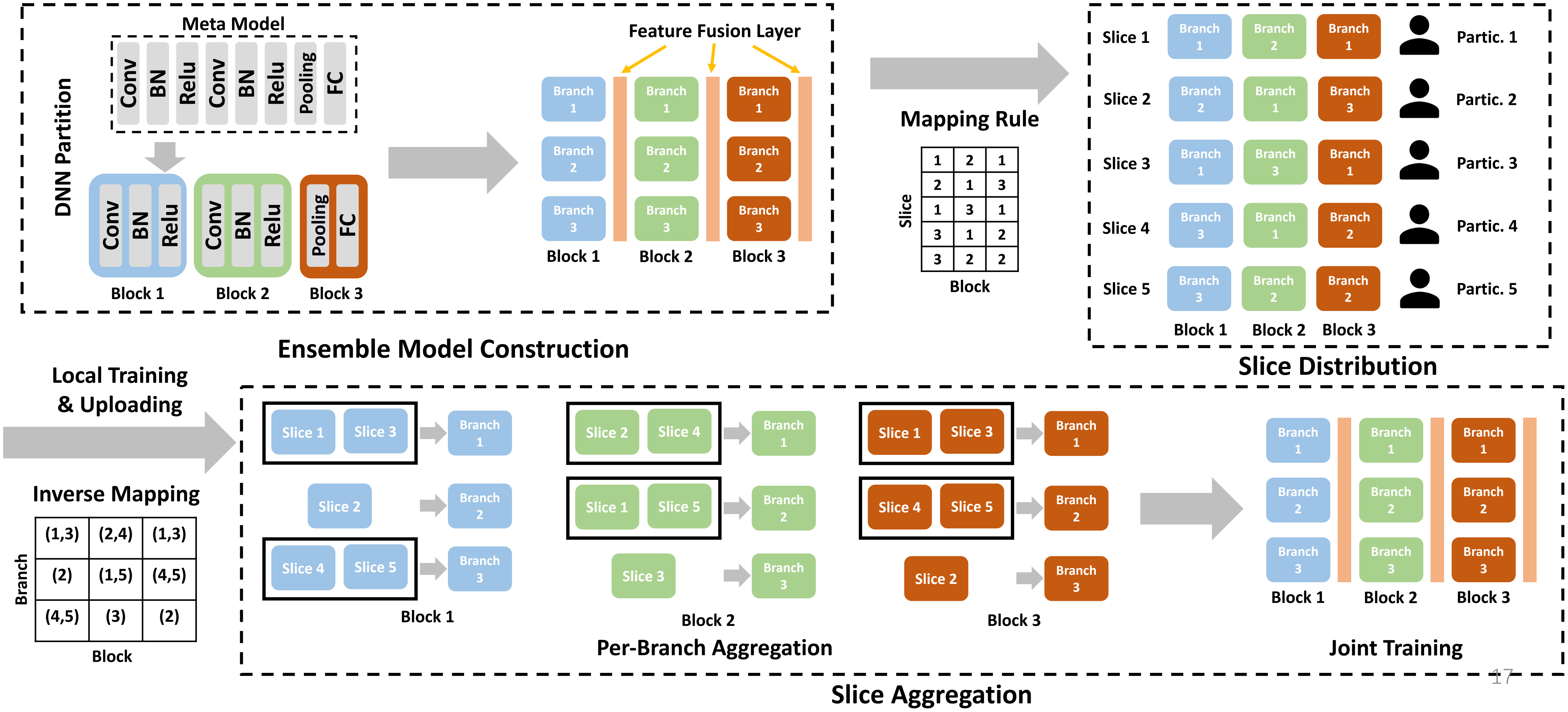


FedSlice Pipeline

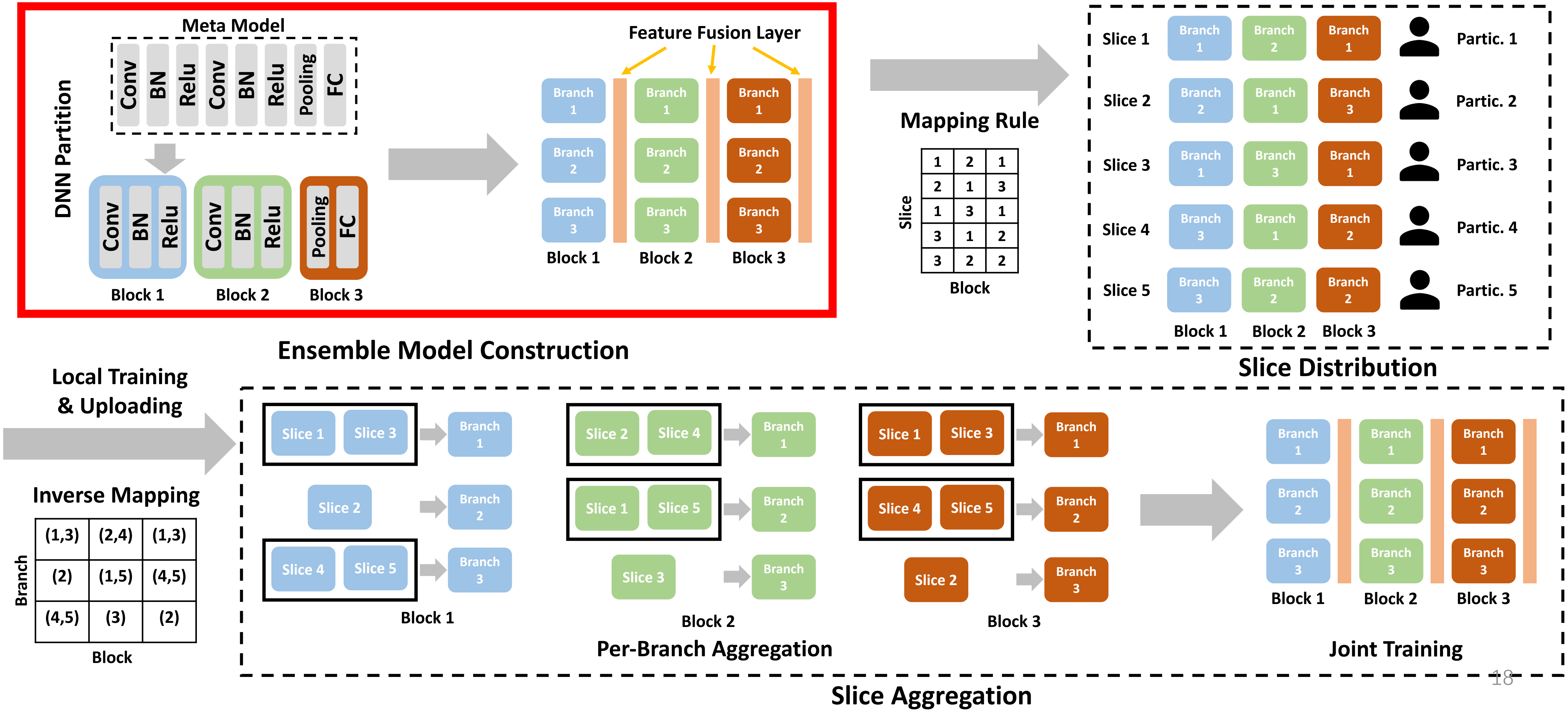




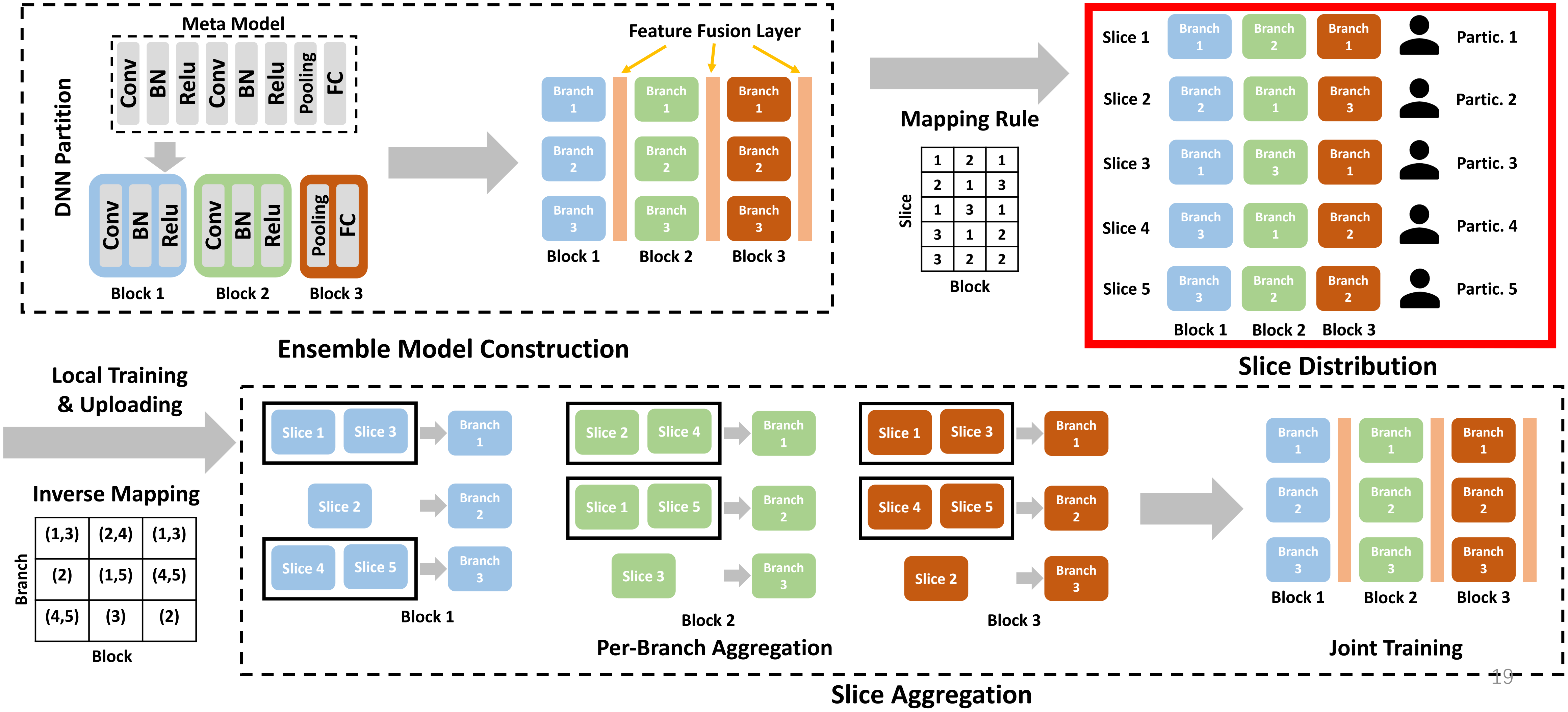
FedSlice Overview



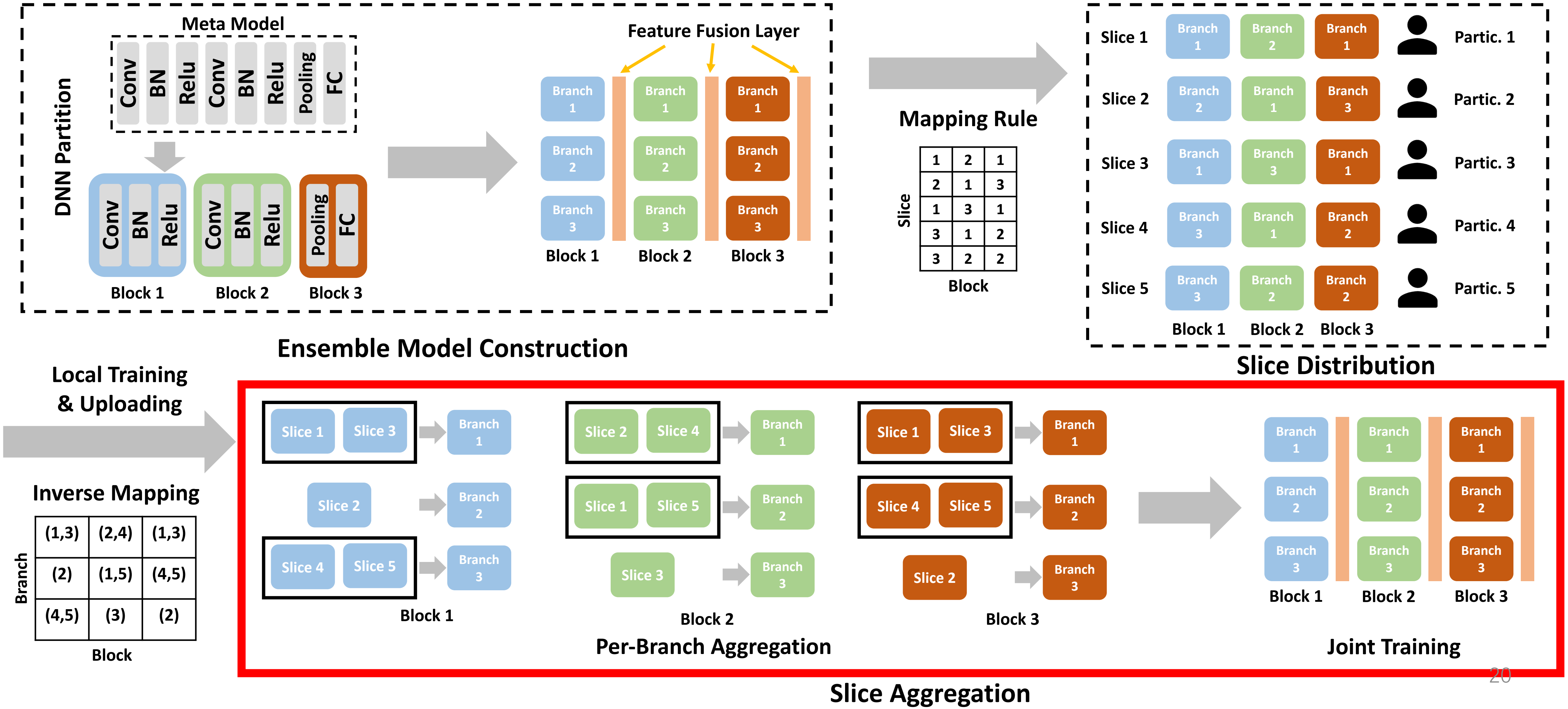
FedSlice Overview



FedSlice Overview



FedSlice Overview



Evaluation Setting

TABLE IV: Description of datasets.

Dataset	Task	#sample	#classes	#partic.	Model
EMNIST	CV	131,500	47	100	CNN
CIFAR10	CV	60,000	10	100	ResNet20
CIFAR100	CV	60,000	100	100	ResNet20
Shakespeare	NLP	517,106	80	143	RNN
FEMNIST	CV	805,263	62	3,550	CNN
Celeba	CV	100,144	2	4,648	ResNet18

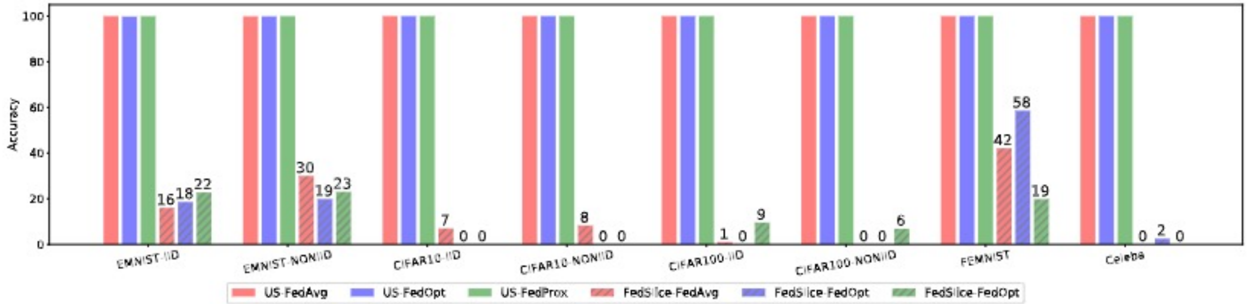
- Dataset
  - Referred to large scale CFL benchmarks: FedML and Leaf
  - Three simulated CV datasets, one NLP dataset, two real-world datasets
- Models
  - Include both CNN and RNN models
- Dataset partition
  - Simulated both IID and non-IID distribution
- Baseline approaches
  - FedAvg, FedProx, FedOpt
- Randomly select several participants to simulate adversary

McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." PMLR, 2017.
Asad, Muhammad, et al. "FedOpt: Towards communication efficiency and privacy preservation in federated learning." Applied Sciences 2020
Li, Tian, et al. “Federated optimization in heterogeneous networks.” MLSys 2020

Defense against Attacks

		McMahan [53]			Asad [6]			Li [43]		
		Baseline	FedSlice		Baseline	FedSlice		Baseline	FedSlice	
			Server ↑	Partic. ↓		Server↑	Partic.↓		Server↑	Partic.↓
EMNIST	IID	79.82	81.74	69.19	79.40	80.65	67.48	83.09	80.19	69.62
	Non-IID	79.04	80.04	61.04	79.44	80.52	59.10	83.04	80.20	69.59
CIFAR10	IID	51.07	50.52	12.87	55.58	51.39	10.36	54.53	51.07	14.48
	Non-IID	48.54	47.20	11.73	50.29	49.05	13.76	52.76	50.04	13.66
CIFAR100	IID	26.50	26.50	1.32	28.54	27.10	1.17	24.21	22.92	7.78
	Non-IID	25.52	26.59	3.7	24.56	25.30	1.13	22.51	23.73	3.93
FEMNIST		73.82	75.02	3.72	72.94	68.66	5.10	76.51	76.34	23.77
Shakespear		41.28	39.52	27.85	42.15	39.95	33.07	39.53	36.72	21.90
Celeba		88.89	86.03	53.73	82.26	84.51	64.75	91.45	89.34	66.98
Average of Relative Value		-	-0.20%	40.63%	-	-1.94%	42.02%	-	-3.11%	47.70%

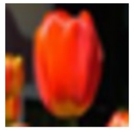
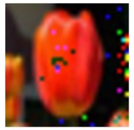
Free-rider attack: 100% to 43%



Adversarial attack: 100% to 11%

		NN				Top3				Loss				Gradient			
		Prec	Recall	F1	Acc	Prec	Recall	F1	Acc	Prec	Recall	F1	Acc	Prec	Recall	F1	Acc
McMahan [53]	US	0.81	0.71	0.69	0.71	0.83	0.76	0.75	0.76	0.81	0.81	0.81	0.81	0.84	0.78	0.77	0.78
	FedSlice	0.52	0.52	0.51	0.52	0.51	0.51	0.51	0.51	0.54	0.53	0.48	0.53	0.53	0.53	0.53	0.53
Asad [6]	US	0.84	0.78	0.77	0.78	0.82	0.73	0.71	0.73	0.85	0.82	0.81	0.82	0.85	0.80	0.79	0.80
	FedSlice	0.50	0.50	0.50	0.50	0.51	0.51	0.50	0.51	0.56	0.55	0.51	0.55	0.55	0.55	0.54	0.55
Li [43]	US	0.59	0.58	0.57	0.58	0.60	0.59	0.58	0.59	0.65	0.64	0.63	0.64	0.63	0.63	0.63	0.63
	FedSlice	0.50	0.50	0.46	0.50	0.50	0.50	0.47	0.50	0.49	0.49	0.38	0.49	0.50	0.50	0.42	0.50
Average	US	0.74	0.69	0.68	0.69	0.75	0.69	0.68	0.69	0.77	0.76	0.75	0.76	0.77	0.74	0.73	0.74
	FedSlice	0.51	0.51	0.49	0.51	0.51	0.51	0.49	0.51	0.53	0.52	0.46	0.52	0.53	0.53	0.50	0.53

Membership inference: downgrade F1 score to random guess

	Ground Truth	Baseline		FedSlice	
		DLG	iDLG	DLG	iDLG
MSE	-	0.00012	0.00014	1.75	1.49
Sample					

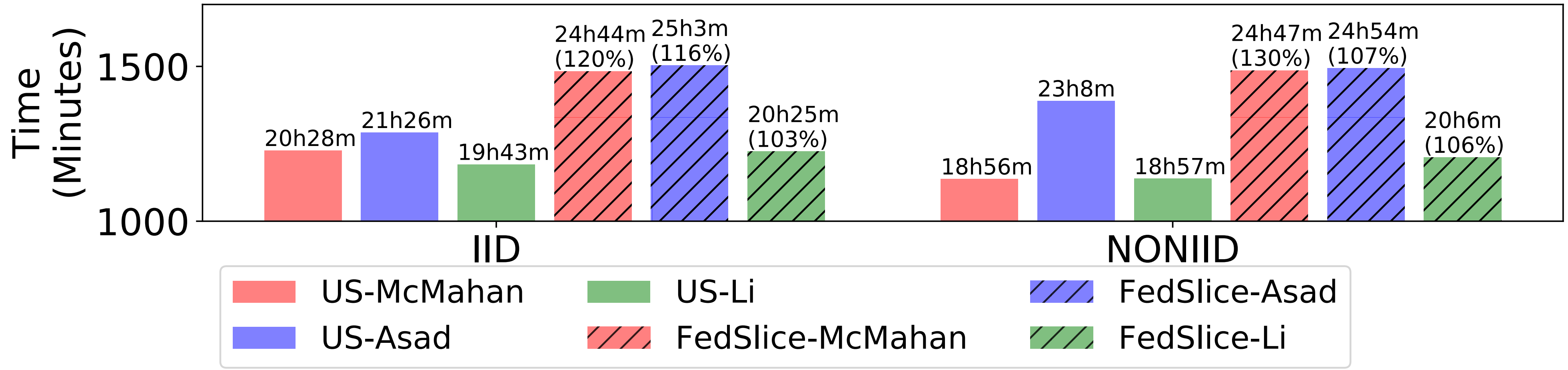
Deep gradient reconstruct: downgrade to random guess

Training Efficiency

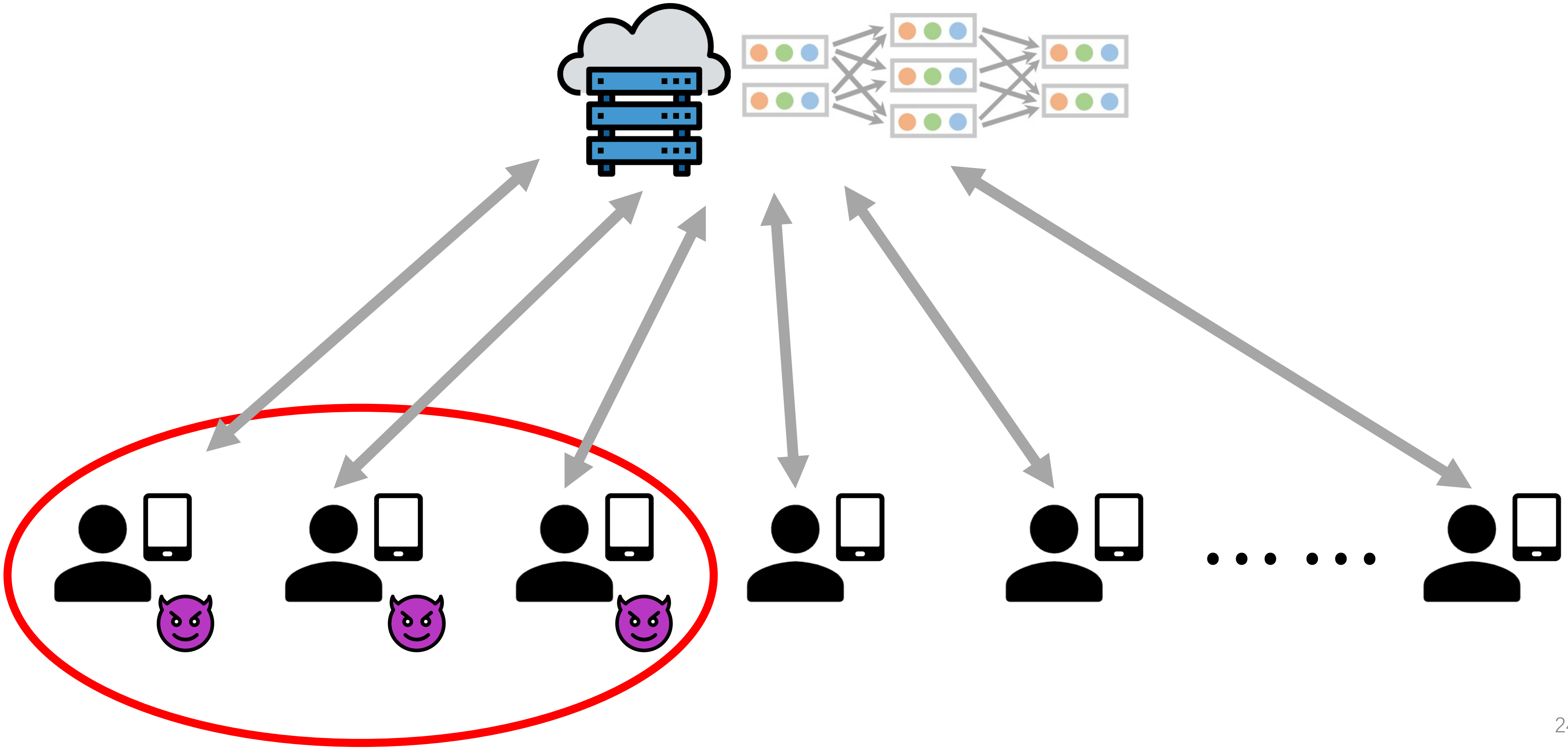
FedSlice takes averagely 14.3% longer time than traditional CFL

31.6X faster than TEE-based solution

877.2X faster than cryptographic solution



Collusion Attack



Collusion Attack

What is the expected number of malicious participants to steal the whole server model?

Coupon Collection Problem

$$
\begin{aligned}
\mathbb{E}(T_2(i,j)) = &\frac{(n-i)\cdot(n-j)}{n\cdot n}[\mathbb{E}(T_2(i,j))+1] \\
&+\frac{(n-i)\cdot j}{n\cdot n}[\mathbb{E}(T_2(i,j-1))+1] \\
&+\frac{i\cdot(n-j)}{n\cdot n}[\mathbb{E}(T_2(i-1,j))+1] \\
&+\frac{i\cdot j}{n\cdot n}[\mathbb{E}(T_2(i-1,j-1))+1].
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}(T_3(0,0,0)) = 0, \\
&\mathbb{E}(T_3(i,0,0)) = \frac{n}{i} + \mathbb{E}(T_3(i-1,0,0)) = n\cdot(\sum_{k=1}^{i}\frac{1}{k}), \\
&\mathbb{E}(T_3(0,j,0)) = \frac{n}{j} + \mathbb{E}(T_3(0,j-1,0)) = n\cdot(\sum_{k=1}^{j}\frac{1}{k}), \\
&\mathbb{E}(T_3(0,0,l)) = \frac{n}{l} + \mathbb{E}(T_3(0,0,l-1)) = n\cdot(\sum_{k=1}^{l}\frac{1}{k}), \\
&\mathbb{E}(T_3(i,j,0)) = \mathbb{E}(T_2(i,j)) \\
&\mathbb{E}(T_3(i,0,l)) = \mathbb{E}(T_2(i,l)) \\
&\mathbb{E}(T_3(0,j,l)) = \mathbb{E}(T_2(j,l))
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}(T_3(i,j,l)) = &\frac{(n-i)\cdot(n-j)\cdot(n-l)}{n^3}[\mathbb{E}(T_3(i,j,0))+1] \\
&+\frac{(n-i)\cdot(n-j)\cdot l}{n^3}[\mathbb{E}(T_3(i,j,l-1))+1] \\
&+\frac{(n-i)\cdot j\cdot(n-l)}{n^3}[\mathbb{E}(T_3(i,j-1,l))+1] \\
&+\frac{i\cdot(n-j)\cdot(n-l)}{n^3}[\mathbb{E}(T_3(i-1,j,l))+1] \\
&+\frac{i\cdot j\cdot(n-l)}{n^3}[\mathbb{E}(T_3(i-1,j-1,l))+1] \\
&+\frac{i\cdot(n-j)\cdot l}{n^3}[\mathbb{E}(T_3(i-1,j,l-1))+1] \\
&+\frac{(n-i)\cdot j\cdot l}{n^3}[\mathbb{E}(T_3(i,j-1,l-1))+1] \\
&+\frac{i\cdot j\cdot l}{n^3}[\mathbb{E}(T_3(i-1,j-1,l-1))+1]
\end{aligned}
$$

Collusion Attack

What is the expected number of malicious participants to steal the whole server model?

Coupon Collection Problem

$$\mathbb{E}(T_3(0,0,0)) = 0,$$

$$\mathbb{E}(T_3(i,0,0)) = \frac{n}{i} + \mathbb{E}(T_3(i-1,0,0)) = n \cdot (\sum_{k=1}^{i} \frac{1}{k}),$$

$$\mathbb{E}(T_3(i,j,0)) = \mathbb{E}(T_2(i,j))$$
$$\mathbb{E}(T_3(i,0,l)) = \mathbb{E}(T_2(i,l))$$
$$\mathbb{E}(T_3(0,j,l)) = \mathbb{E}(T_2(j,l))$$

$$\begin{aligned}\mathbb{E}(T_3(i,j,l)) = &\frac{(n-i)\cdot(n-j)\cdot(n-l)}{n^3}[\mathbb{E}(T_3(i,j,0)) + 1] \\ &+ \frac{(n-i)\cdot(n-j)\cdot l}{n^3}[\mathbb{E}(T_3(i,j,l-1)) + 1] \\ &+ \frac{(n-i)\cdot j\cdot(n-l)}{n^3}[\ldots] \\ &\ldots \\ &+ \frac{(n-i)\cdot j\cdot l}{n^3}[\mathbb{E}(T_3(i,j-1,l-1)) + 1] \\ &+ \frac{i\cdot j\cdot l}{n^3}[\mathbb{E}(T_3(i-1,j-1,l-1)) + 1]\end{aligned}$$

The ratio of expected number of colluded participants is 38.85%

Conclusion

Motivated from traditional program slicing, we employ model slicing to solve the security problems in CFL

We propose a slice-based framework, FedSlice, to simultaneous defense four attacks initiated from malicious participants

We conduct extensive experiments with real-world datasets to demonstrate the effectiveness, efficiency, and scalability of FedSlice

Thanks for Listening

Q & A