PCAT: Functionality and Data Stealing from Split Learning by Pseudo-Client Attack

USENIX Security 23

Xinben Gao   Lan Zhang*

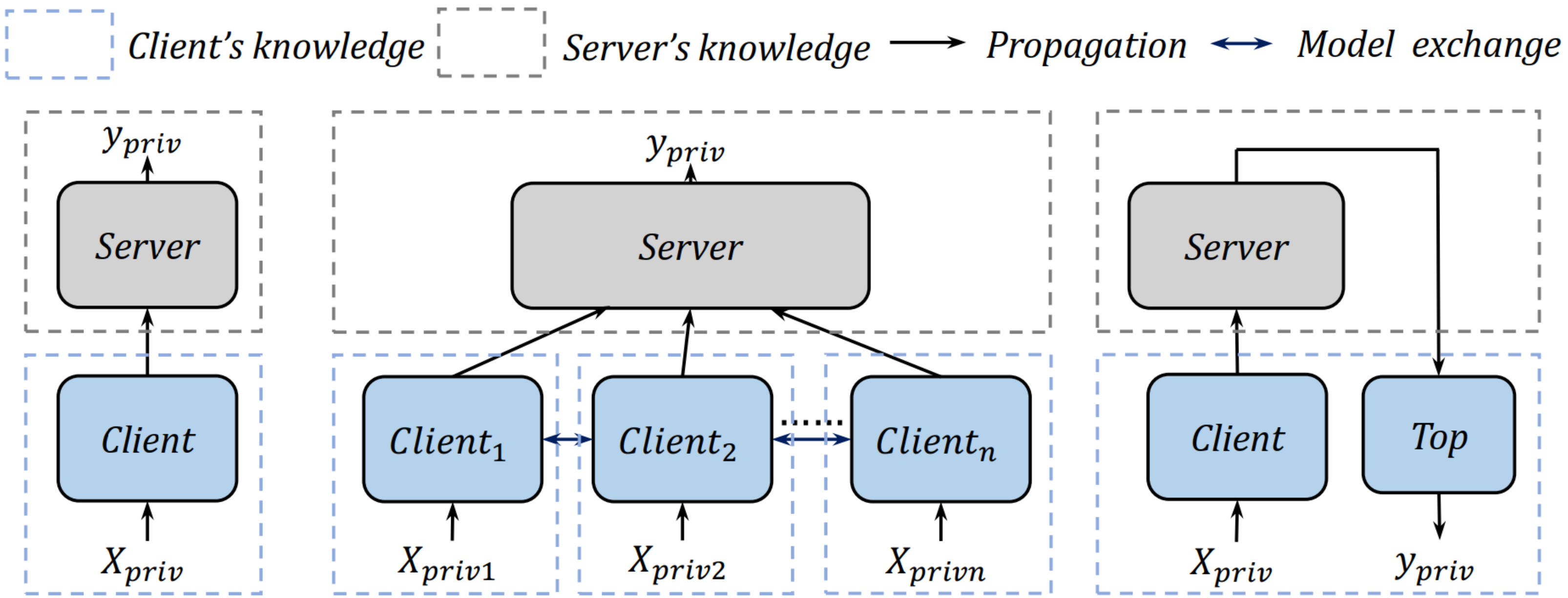
University of Science and Technology of China

Lab for Intelligent Network and Knowledge Engineering

Content

Background: Split learning (SL)

A paradigm of distributed ML.
Design for protecting the client's privacy.



(a) Two-part single-client (b) Two-part multi-client (c) U-Shape

Is there any risk of leaking private information?

Background: Previous Work

	FSHA[1]	UnSplit[2]	PCAT(Ours)
Attack	Malicious	Semi-honest	Semi-honest
Functionality Stealing	×	√	√
Input reconstruction	√	√	√
Label inference	×	√	√
Suit complex case	√	×	√

[1] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. (CCS2021)

[2] Ege Erdogan, Alptekin Küpçü, and A. Ercüment Çiçek. Unsplit: Data-oblivious model inversion, model stealing, and label inference attacks against split learning. (WPES@CCS 2022)

Attack Goals

More general and challenging scenario:

Transparent to the client

Minimal knowledge about the client model

Support more complex client models and tasks

Effective against three variants of SL

Resilient to some defensive methods

Assumption

The server has a tiny dataset for the same learning task

Content

Insight

Model trained on a small dataset (attack model)

Steal Functionality →

Model trained on a large dataset (victim model)

scenarios

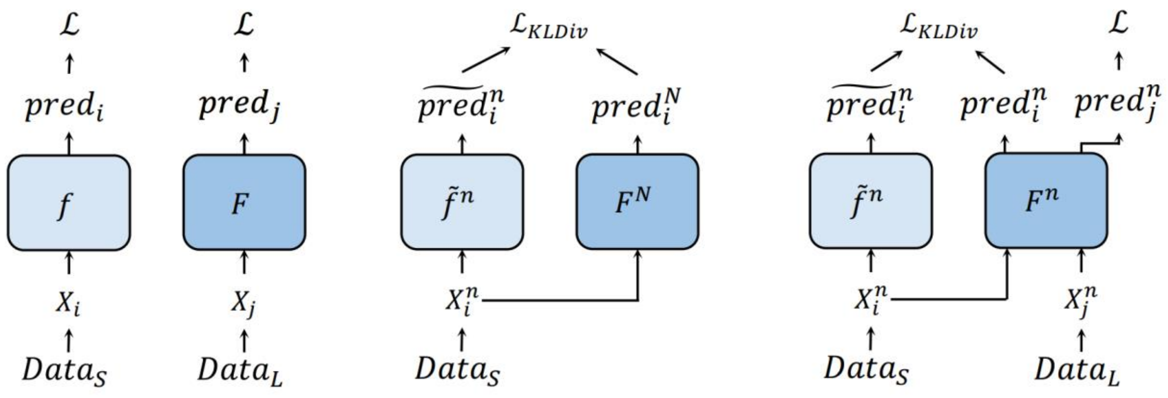
1. Stealing a complete model
2. Stealing a client model

strategies

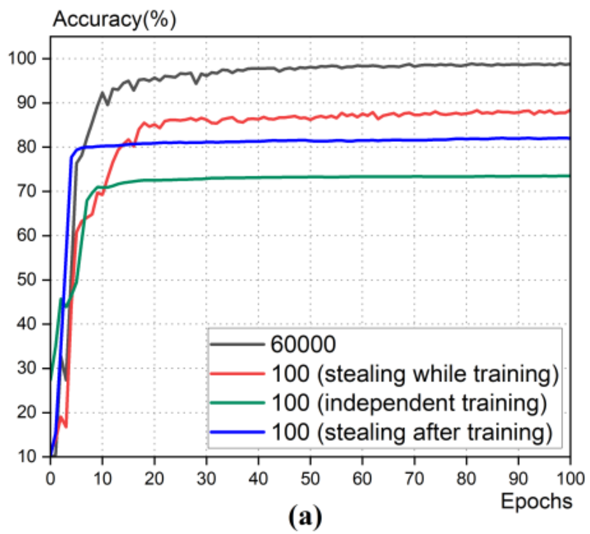
1. Stealing after training
2. Stealing while training

Insight: Steal a complete model

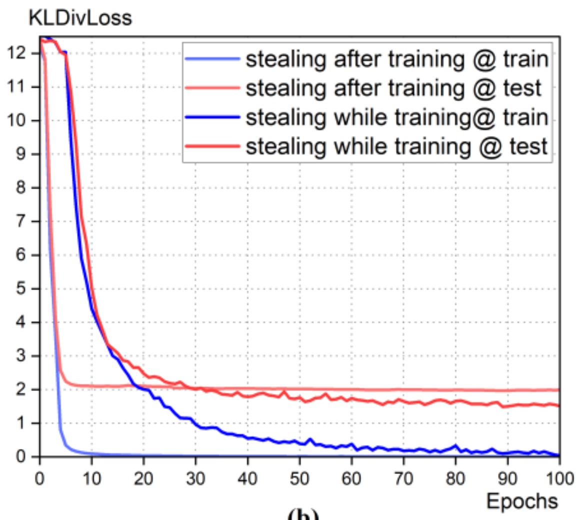
The evolving learning targets can "guide" the attack model to converge more precisely to the victim model.



(a) Independent training (b) Stealing after training (c) Stealing while training



(a)



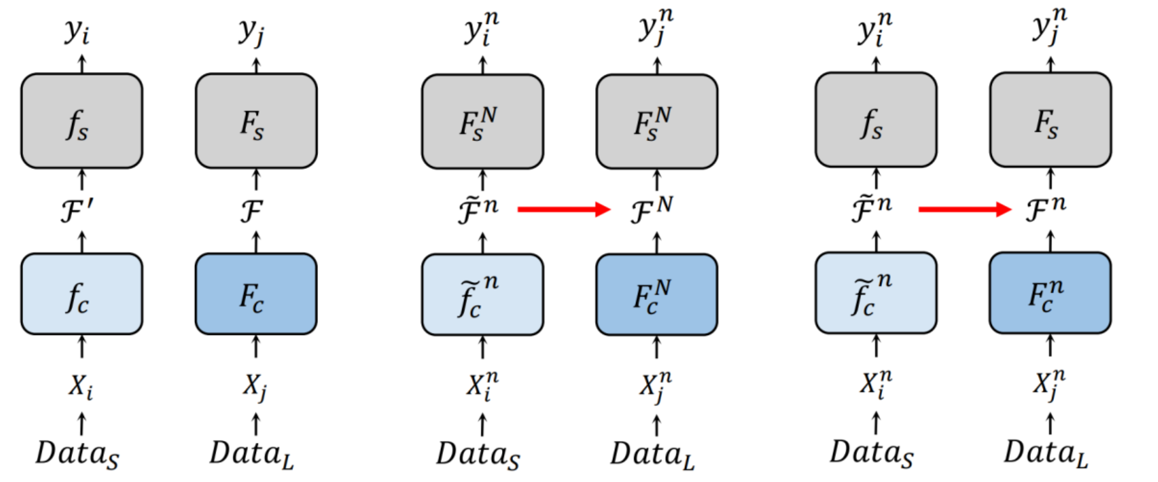
(b)

Insight: Steal a client model

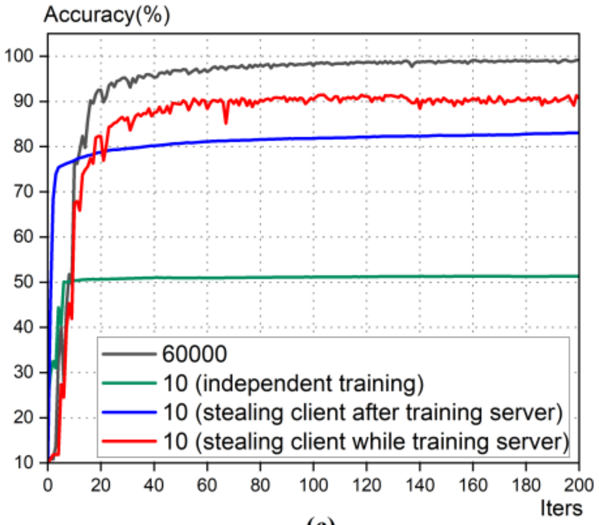
Challenge:

1. The attack client can't obtain the victim client, it only obtain the server model.

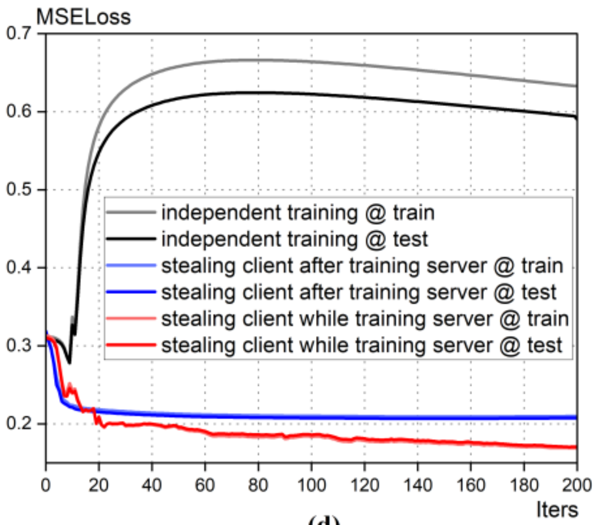
2. The attack client can't feed data to the victim client and get soft labels generated by the victim client.



(a) Independent training (b) Stealing after training (c) Stealing while training



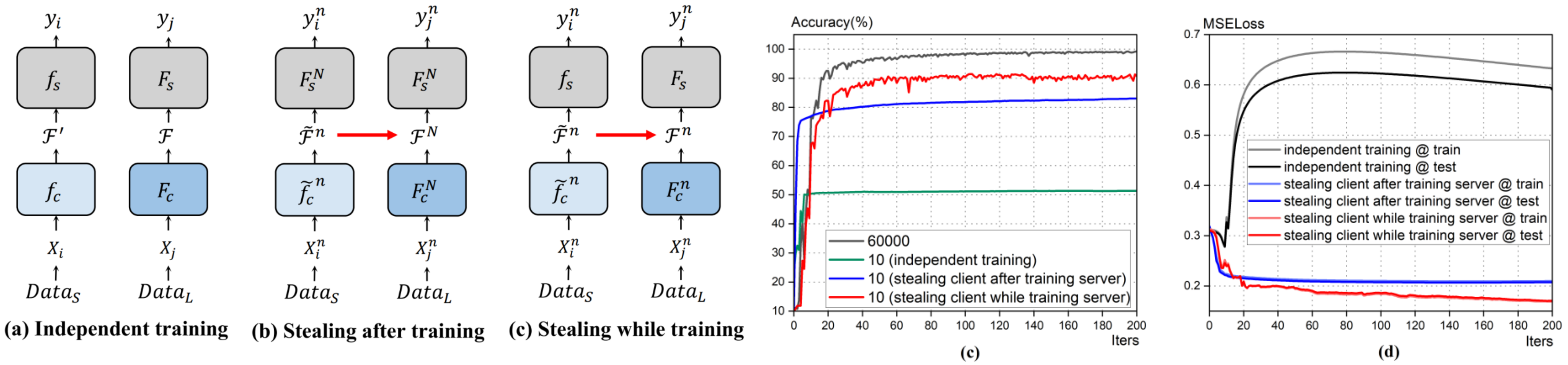
(c)



(d)

Insight: Steal a client model

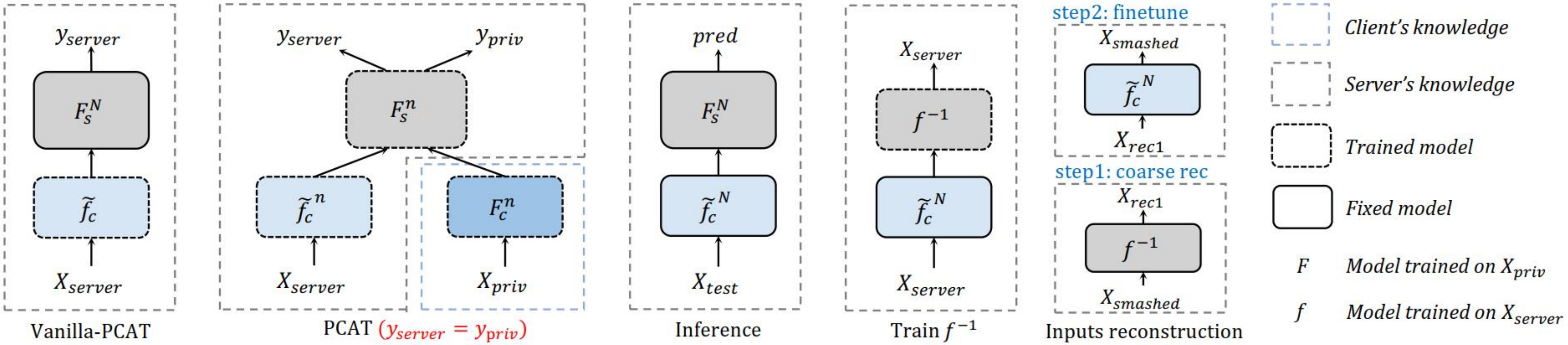
The attack client optimizes the feature space of its output to get closed to the feature space of the victim client's output.



Content

Pseudo-client Attack (PCAT)

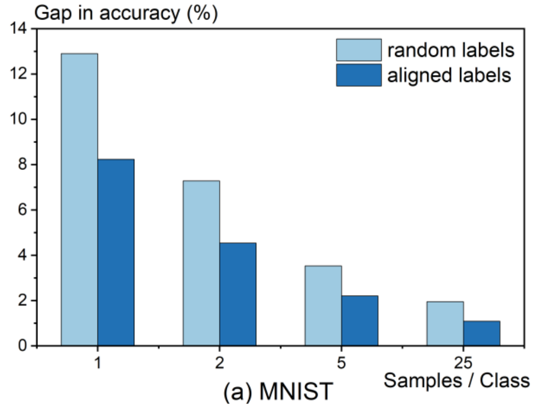
- ❑ **Steal functionality**
- ❑ **Perform inference alone**
- ❑ **Train reverse mapping**
- ❑ **Reconstruct private inputs**

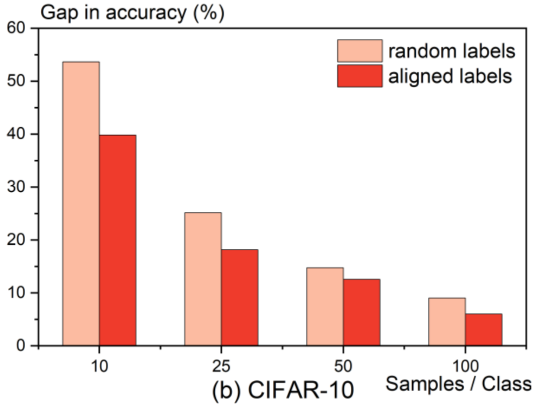


Details of PCAT

Aligning labels

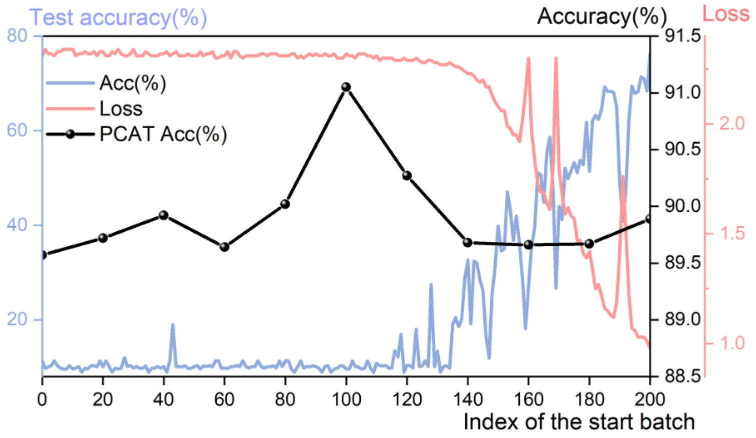
$$y_{server} = y_{priv}$$





Late start

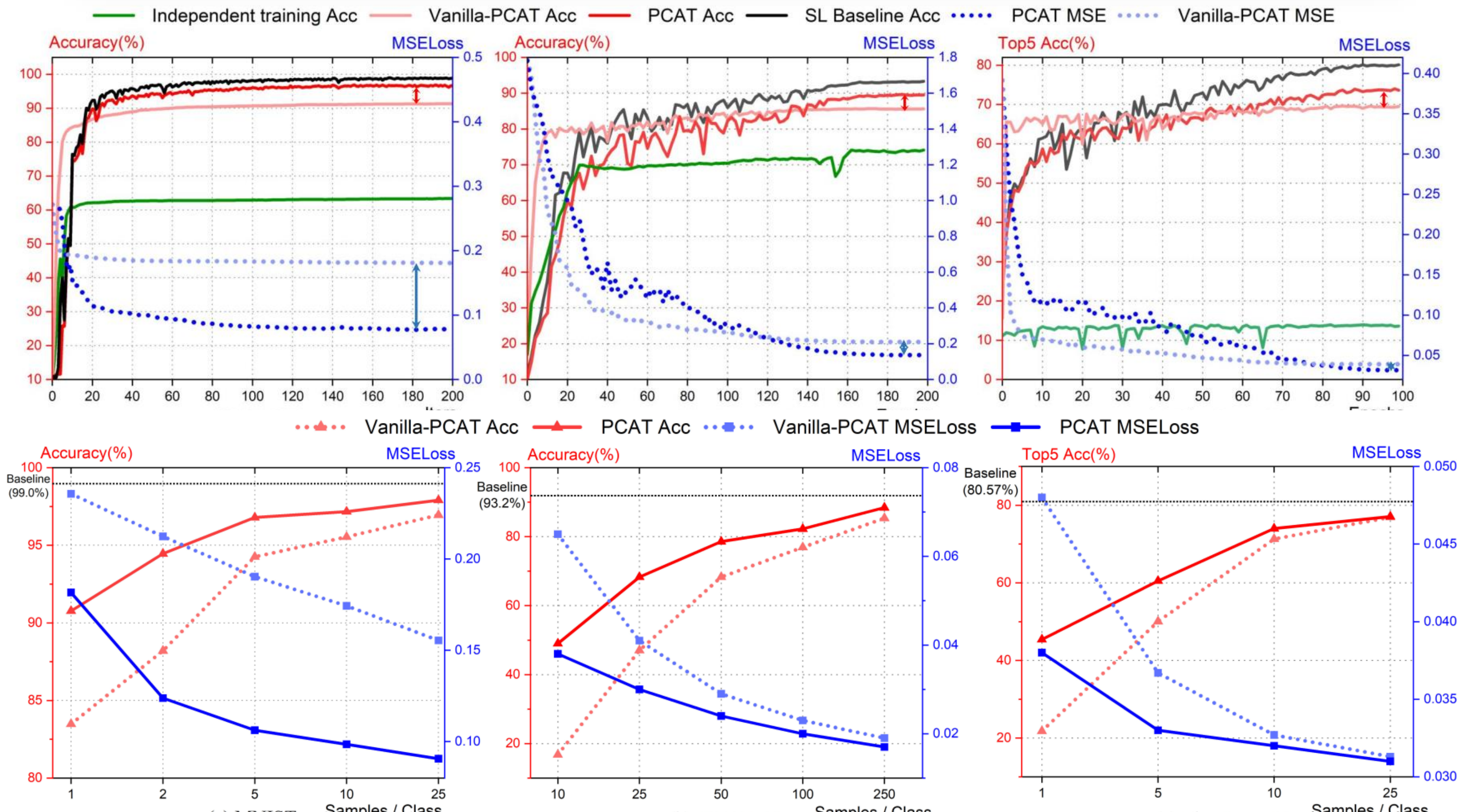
Skip some iterations at the beginning epochs



Content

Experiment results

Functionality stealing result on
MNIST, CIFAR-10 and Tiny-Imagenet

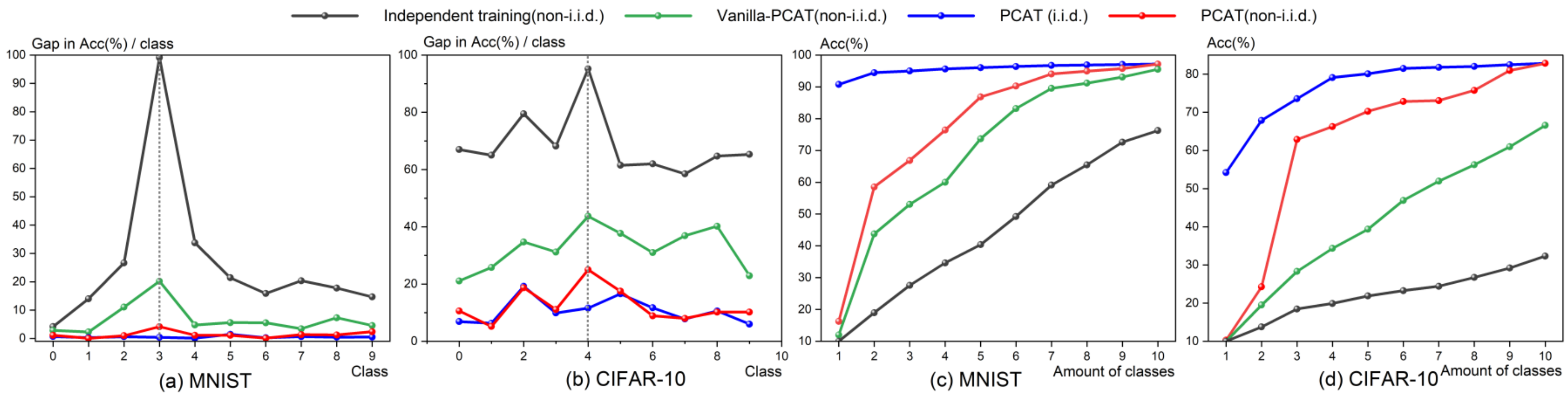


(a) MNIST (b) CIFAR-10 (c) Tiny ImageNet

Experiment results

Functionality stealing result on non-i.i.d. dataset.

PCAT is **robust** to non-i.i.d. cases.



Experiment results

PCAT performs well though the server model and the victim model is different.

	Pseudo client			Victim client
	Simple	Same	Complex	
Model	MaxPool ReLU Conv2d	MaxPool ReLU Conv2d / MaxPool ReLU Conv2d	MaxPool ReLU Conv2d / ReLU Conv2d / MaxPool ReLU Conv2d	MaxPool ReLU Conv2d / MaxPool ReLU Conv2d
Acc(%)	73.60	97.17	97.13	99.06
MSE	0.387	0.133	0.141	0

	Pseudo client				Victim client
	Simple	Same	Complex	Other	
Model	MaxPool Conv2d / MaxPool Conv2d	MaxPool Conv2d Conv2d / MaxPool Conv2d Conv2d	MaxPool Conv2d Conv2d Conv2d / MaxPool Conv2d Conv2d Conv2d	ResBlock / ResBlock	MaxPool Conv2d Conv2d / MaxPool Conv2d Conv2d
Acc(%)	87.54	88.90	88.35	84.96	93.20
MSE	0.0279	0.0134	0.0166	0.0511	0

Experiment results

Our attack is resilient to privacy defenses the victim clients may adopts.

NoPeek defense

MNIST					
α	0	0.2	0.4	0.6	0.8
Baseline Acc(%)	99.00	98.52	98.10	96.98	94.33
PCAT Acc(%)	98.01	97.27	96.89	93.41	92.55
Acc(%) Gap	0.99	1.25	1.21	3.57	1.78
CIFAR-10					
α	0	0.1	0.2	0.4	0.6
Baseline Acc(%)	93.2	87.56	78.64	68.04	62.61
PCAT Acc(%)	82.77	75.29	64.42	60.05	55.13
Acc(%) Gap	10.43	12.27	14.22	7.99	7.47

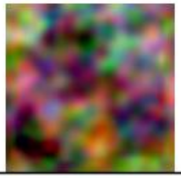
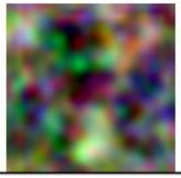
DP-noise on the client model

MNIST				
σ	+∞	70	60	50
Baseline Acc(%)	99.00	94.10	90.79	84.71
PCAT Acc(%)	97.31	91.12	88.66	80.84
Acc(%) Gap	1.69	2.98	2.13	3.87
CIFAR-10				
σ	+∞	200	100	50
Baseline Acc(%)	93.20	85.18	80.17	73.17
PCAT Acc(%)	86.50	77.45	71.14	68.34
Acc(%) Gap	6.70	7.73	9.03	4.83

Experiment results

Appropriate Gaussian noise to the smashed data can improve attack performance

DP-noise on smashed data

σ	0	0.1	0.3	0.5
Baseline Acc(%)	80.28	79.80	79.90	80.07
PCAT Acc(%)	74.52	77.79	79.00	79.45
MSE	0.0362	0.0864	0.2108	0.3690
				

Experiment results

Our attack outperforms SOTA method in every attack goals.

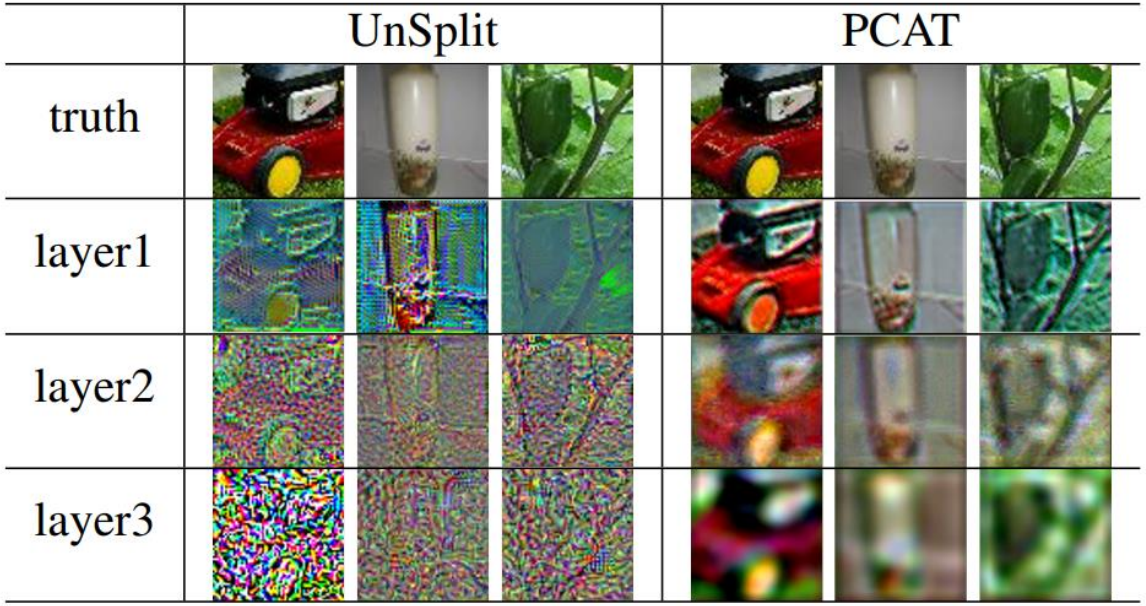
Functionality stealing

Datasets	MNIST		CIFAR-10	
Methods	UnSplit [9]	PCAT	UnSplit [9]	PCAT
SL Baseline	98.00	99.00	71.00	93.20
split layer = 1	93.75	**98.75**	43.69	**91.10**
split layer = 2	63.3	**96.79**	22.12	**78.57**

Label inference

Datasets	MNIST		CIFAR-10	
Methods	UnSplit	PCAT	UnSplit	PCAT
top layer = 1	100.0	**98.82**	100.0	**93.42**
top layer = 2	9.1	**96.58**	8.1	**92.57**

Data reconstruction



Content

Conclusion

A novel attack

Applicable on various split learning settings

Achieve several attack goals

Unknown victim client model

Works effectively for rich models, tasks and settings

Transparent to the client

Q & A

Thank you!

Please feel free to contact with us:

Xinben Gao: gxb1320276347@mail.ustc.edu.cn

Lan Zhang: zhanglan@ustc.edu.cn