Federated Analytics: A New Collaborative Computing Paradigm towards Privacy Focusing World

Presenter, Dan Wang and Siping Shi

Department of Computing, The Hong Kong Polytechnic University

Thanks to: Chuang Hu, The Hong Kong Polytechnic University,
Zibo Wang, Yifei Zhu from Shanghai Jiaotong University,
Dawei Chen, Yuhan Kang, Zhu Han, University of Houston

Tutorial outline

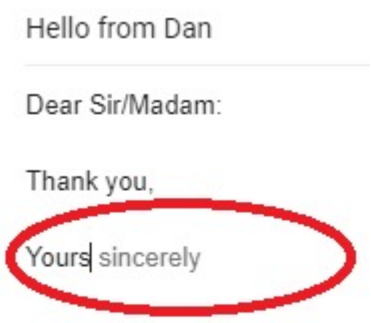
- Introduction to Federated Analytics
  - A brief background on federated learning
  - What is Federated Analytics and Why Federated Analytics
  - Federated video analytics: a first example on federated analytics
- Opportunities and challenges
- Federated analytics examples
  - Federated analytics for privacy-demanding systems
  - Federated analytics for enhancing privacy-preserving systems
- Conclusions

A brief background on federated learning

A brief background on federated learning

- Introduced by Google in 2017
  - Gboard application
  - To collaboratively train a machine learning model where the data are kept local

- Two initial goals
  - To reduce the amount of communications
  - To handle the privacy concerns



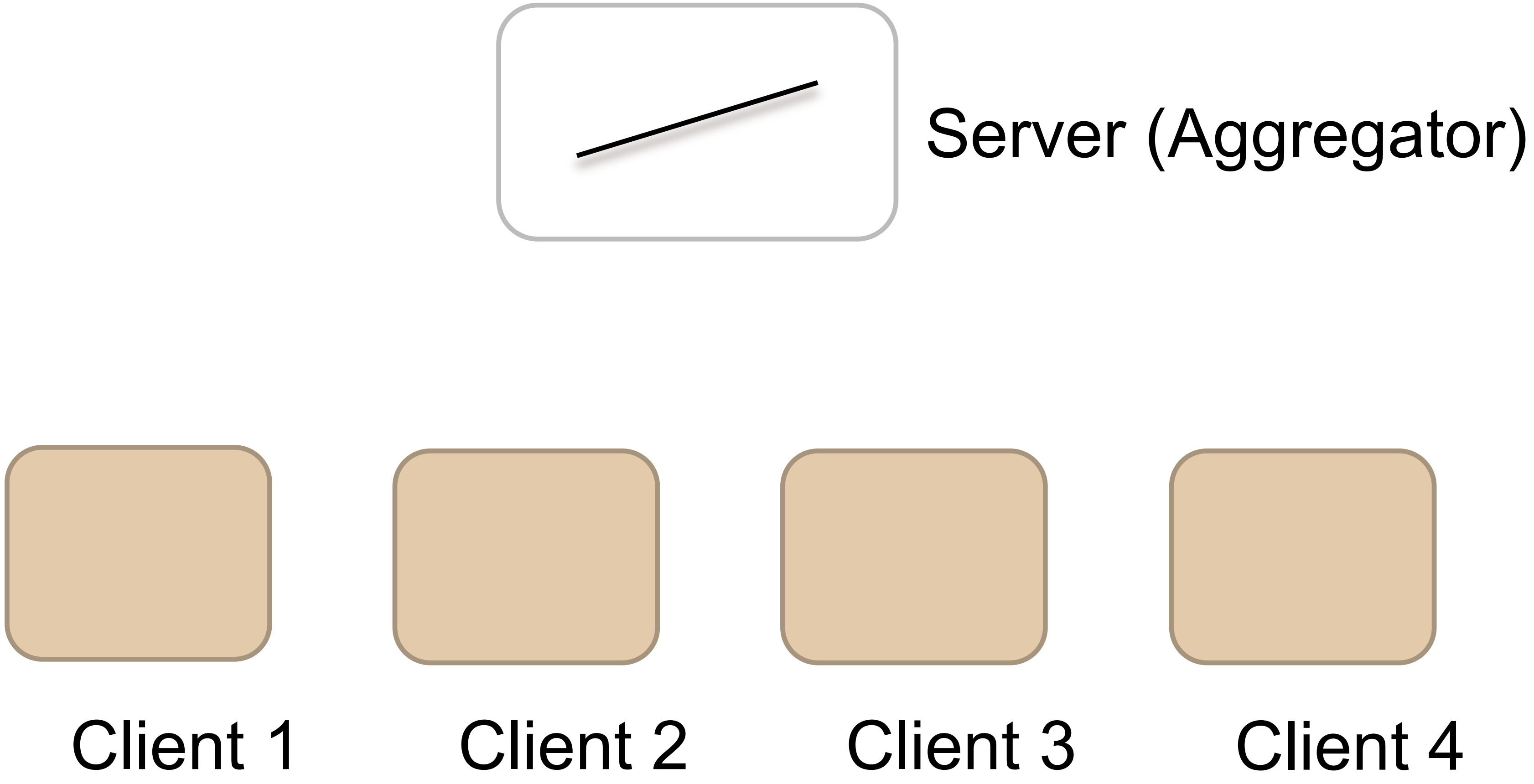
Hello from Dan

Dear Sir/Madam:

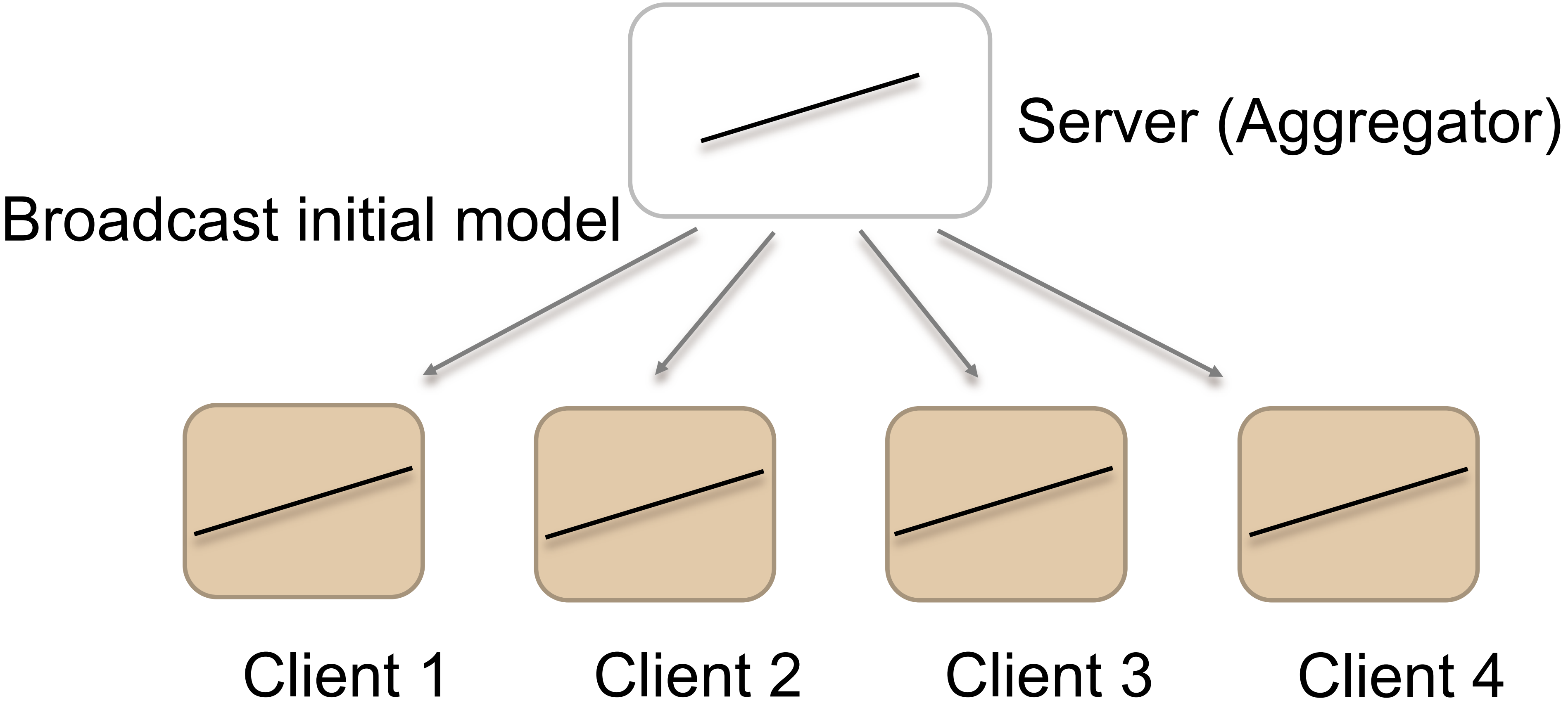
Thank you,

Yours sincerely

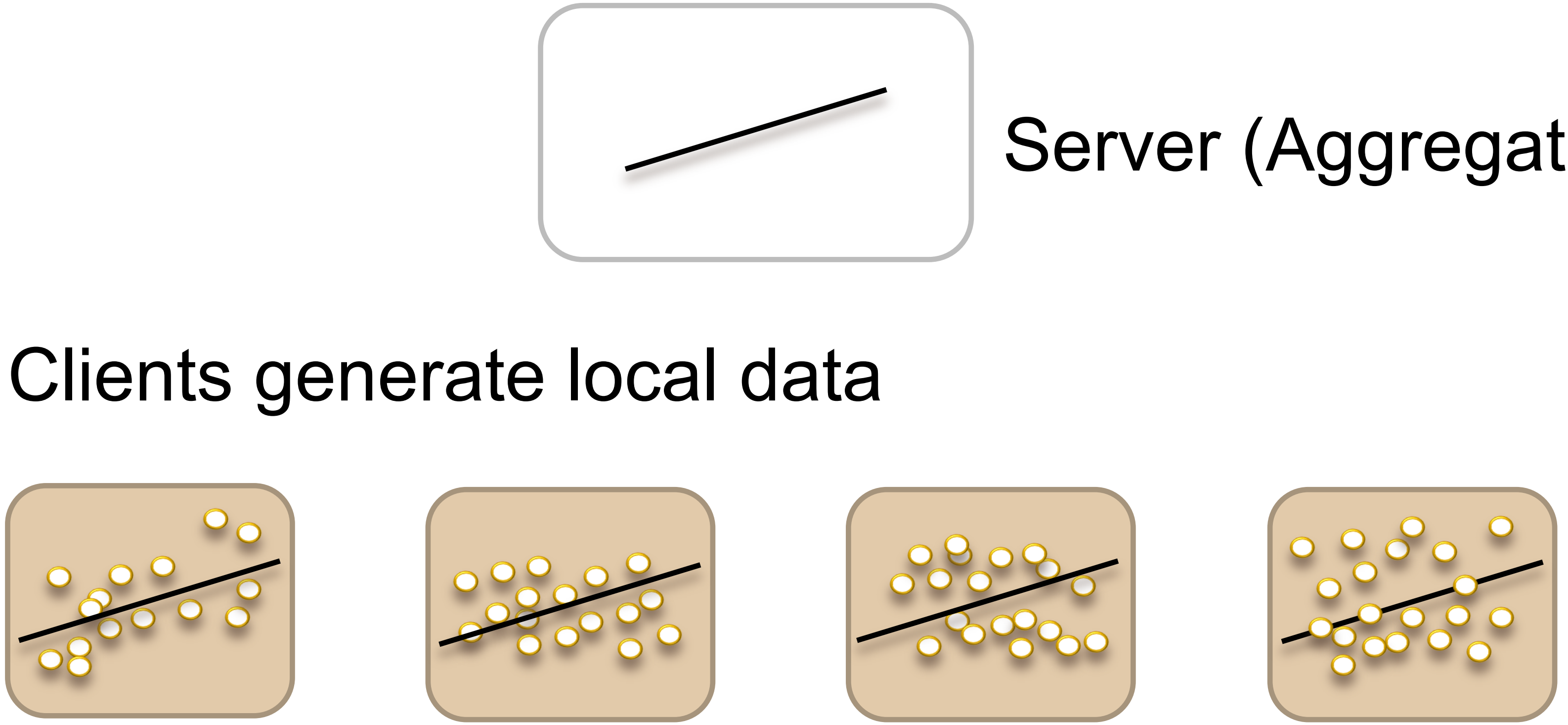
Federated learning in operation



Federated learning in operation



Federated learning in operation

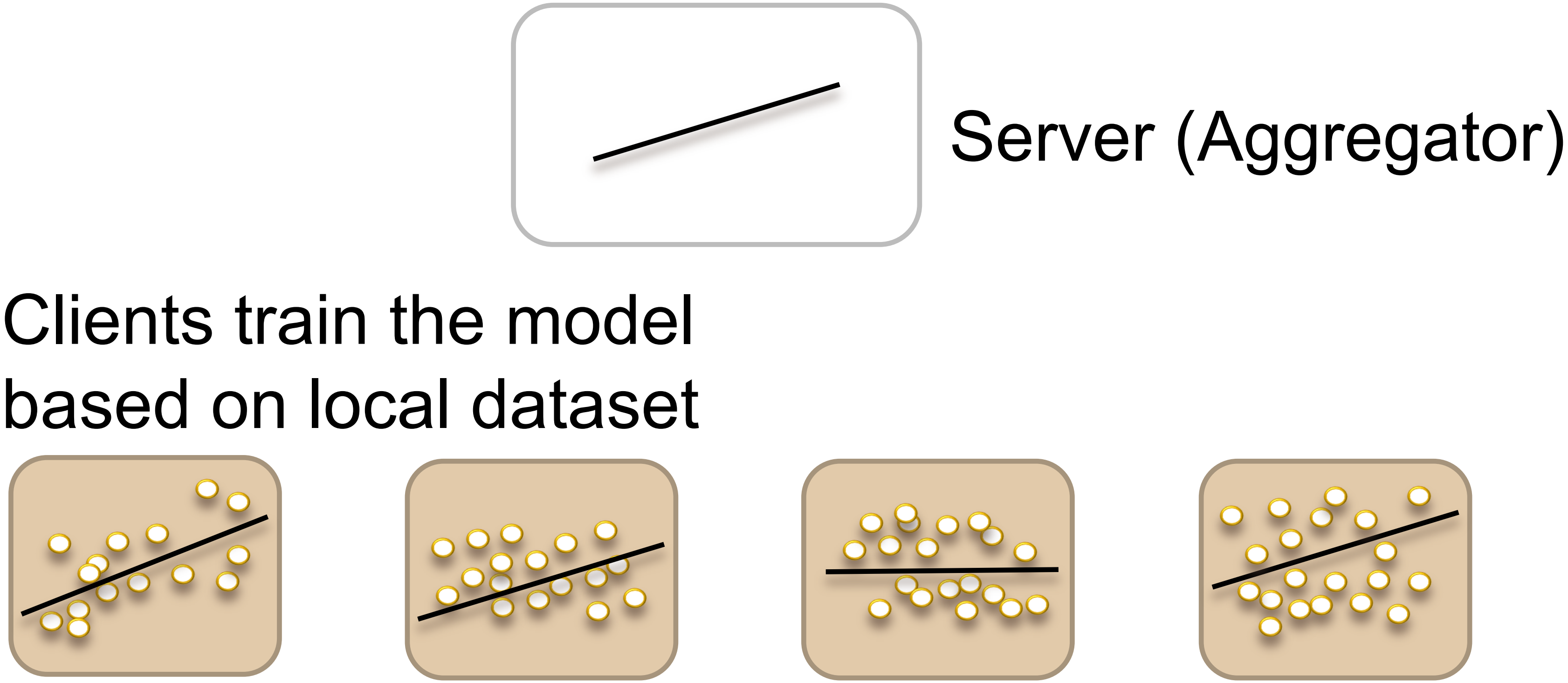


Server (Aggregator)

Clients generate local data

Client 1 Client 2 Client 3 Client 4

Federated learning in operation



Server (Aggregator)

Clients train the model based on local dataset

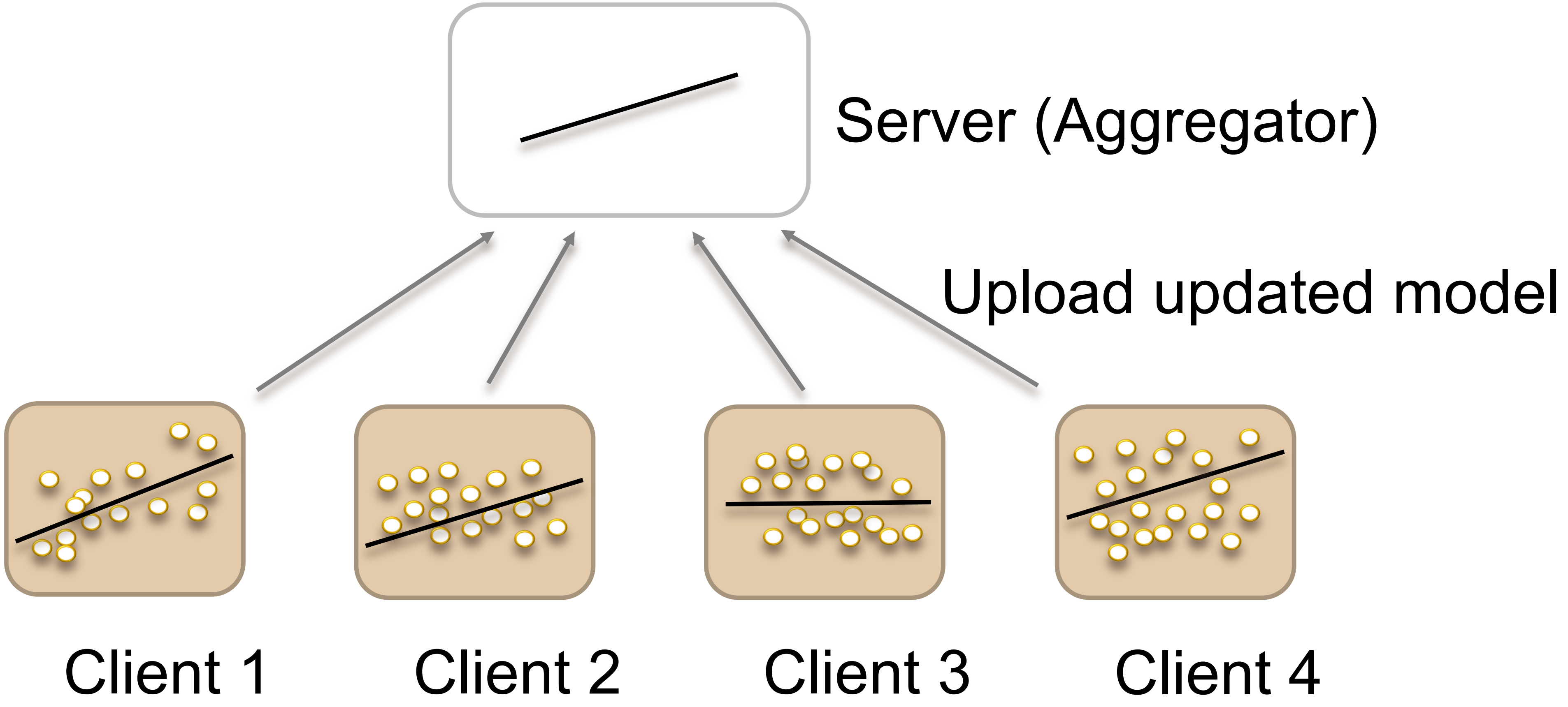
Client 1

Client 2

Client 3

Client 4

Federated learning in operation

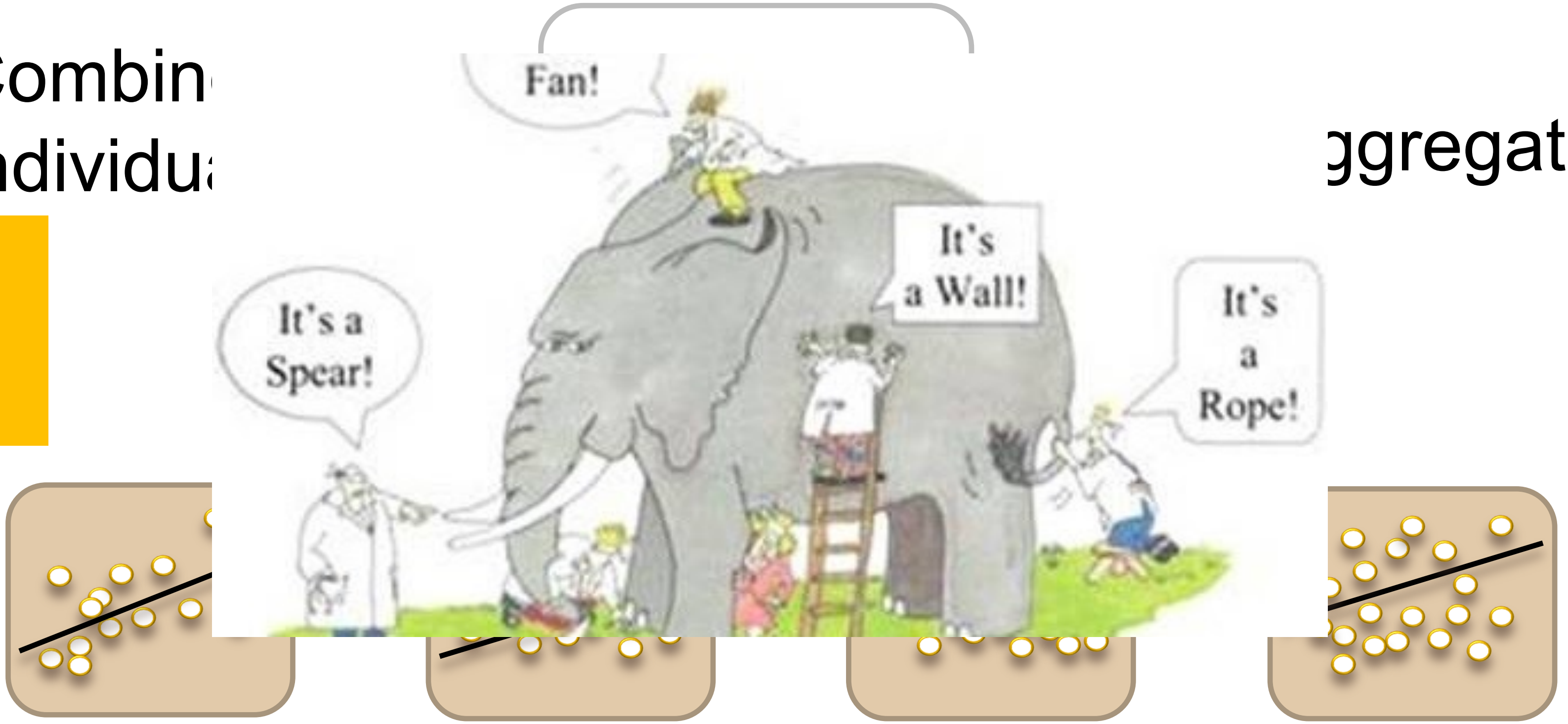


Federated learning in operation

Combin
individu

ggregator)

Repeat these process until convergence



Client 1

Client 2

Client 3

Client 4

Current status on federated learning

- The focus is on privacy

- Cross-device federated learning
  - Networking and systems community
    - Usually emphasizes on large scale, model training of neural network models
  - Distributed computing, optimization and algorithm community
    - Usually emphasizes on distributed optimization, algorithms, sometimes on non-neural network models

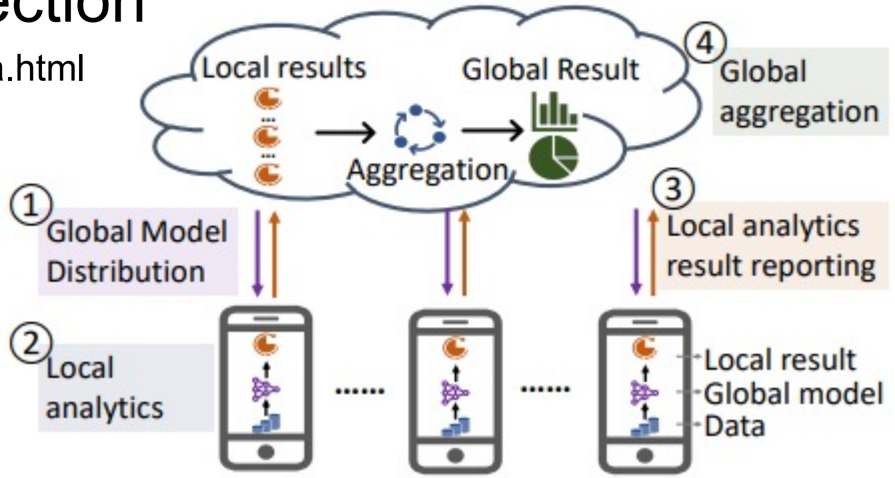
- Cross-silo federated learning
  - Data mining community
    - Usually emphasizes on different domains, to solve the data island problem, finance, medical, etc.

What is federated analytics and Why federated analytics

What is Federated Analytics

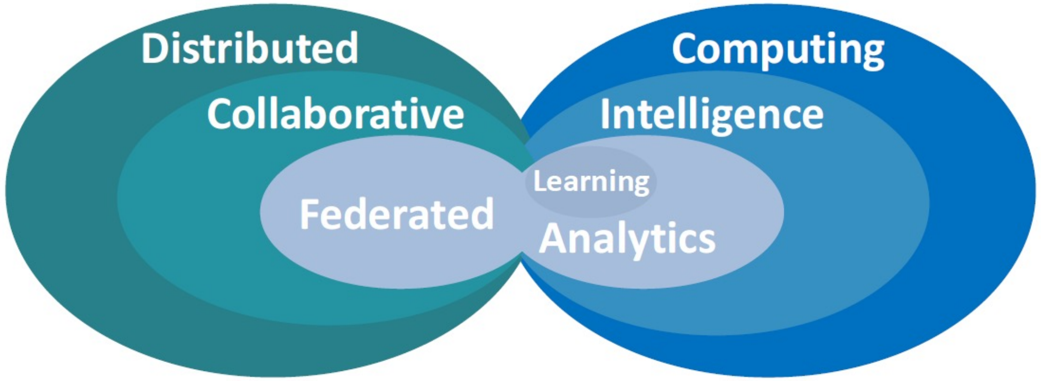
- Google proposed Federated Analytics in May 2020
  - Also for the Gboard application
  - Federated learning for model training
  - Federated analytics for model testing

- Google's definition on federated analytics: Collaborative data science without data collection
  - https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html



What is Federated Analytics: Taxonomy

- Federated: how nodes collaborate
- Analytics: what the computing task is



Collaboration Model Computing Model

- Data analytics: to draw conclusions from data
- Federated Analytics: A collaborative computing paradigm that performs data analytic computing tasks across multiple decentalized devices where the raw data should be kept local

Wang, Shi, Zhu, Han, “Federated Analytics: Opportunities and Challenges”, to appear, *IEEE Network*, 2021

Federated Analytics vs. others

- To Federated Learning

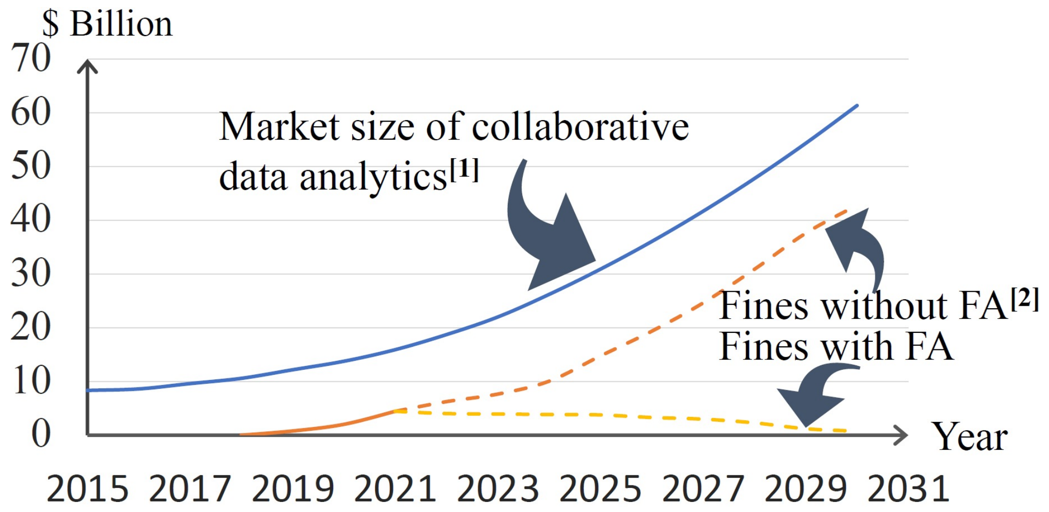
	Federated Learning	Federated Analytics
Goal	Training ML models	Non-training tasks (data science)
Aggregation approach	FedAvg, etc	Task dependent Tree | Bayesian | MPC | etc.
Local insights	Model weights	Task dependent Partial info | Distilled info | etc.

- To Distributed (Data) Analytics

	Distributed (Data) Analytics	Federated Analytics
Raw data transmission	Redistribution assumed	Stay where it origins
Clients (nodes) and server	Trusted	Untrusted (privacy & Byzantine attack)
Data & device heterogeneity	Little concerned	Focused

Why Federated Analytics

- Application demands
  - Increasing demands on collaborative data analytics
  - Increasing concerns on privacy and confidentiality
    - Google fined USD$57 million, 2019



[1] Advanced Analytics Market : https://www.abnewswire.com/pressreleases/advanced-analytics-market-2019-2023-business-trends-emerging-technologies-size-global-segments-and-industry-profit-growth_436058.html
[2] GDPR Enforcement Tracker : https://www.enforcementtracker.com/

Why Federated Analytics

- Resource readiness
  - Increasing data generation in the edge
  - Increasing edge resources

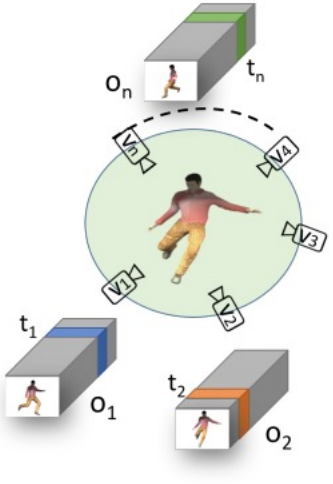
- Technology readiness
  - Platform development, e.g., Tensorflow Federated
  - Edge technologies, e.g., edge-cloud computing, serverless computing, edge acceleration
  - Analytics technologies: data analytics, video analytics
  - Computational privacy metrics and technologies, e.g., Differential Privacy, Homomorphic Encryption, Secure Multi-party Computation

Federated video analytics: a first example

Federated video analytics: a first example

- Video analytics recognize/establish spatial or temporal events/environment from video frames

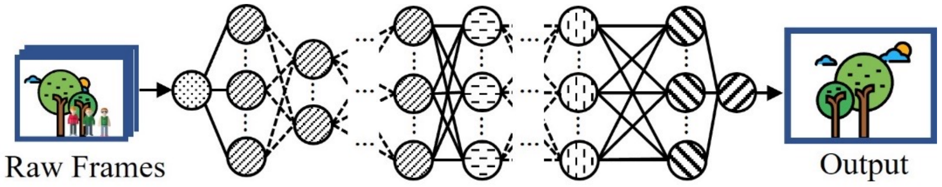
- Multi-camera 3D construction:
  - High-definition maps, digital twins, metaverse
  - Improve limited view scope, low resolution, image missing or errors



Hu, Lu, Wang, "FEVA: A FEderated Video Analytics Architecture for Networked Smart Cameras", to appear, *IEEE Network, special issue on Interplay Between Machine Learning and Networking Systems,* 2021

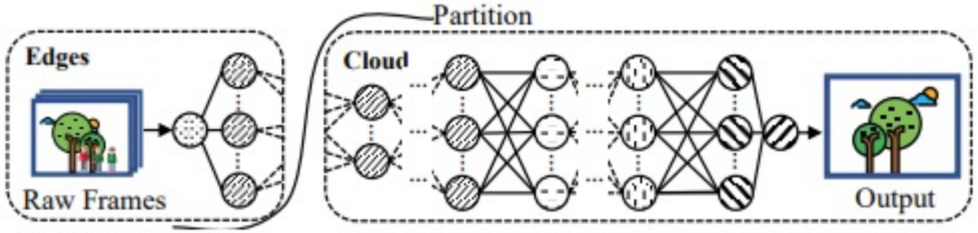
Background on video analytics

- The general computing pipeline of video analytics
  - Video frames are fed into a pre-trained neural network model to output

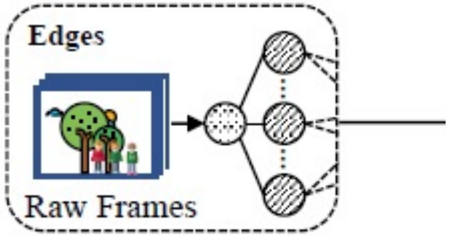


A pre-trained neural network model for a 3D reconstruction task

- Video analytics through edge-cloud computing
  - Real-time responses
  - Edge has resource constraints
  - Resource-aware optimization/partition

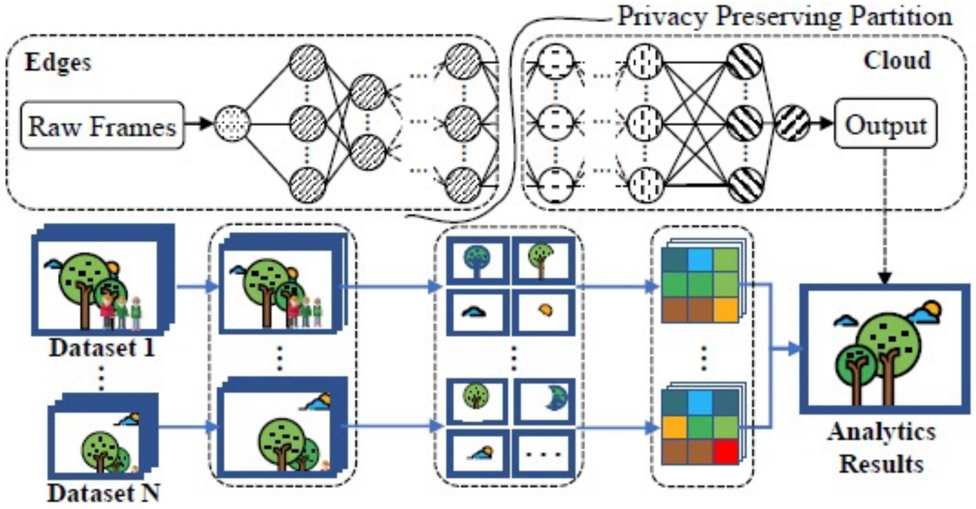


The potential privacy problem



Privacy-sensitive Analytics-irrelevant information

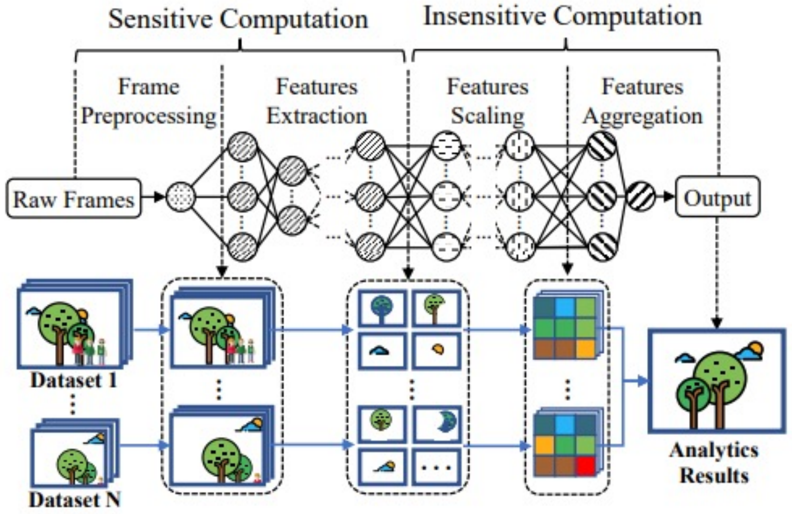
- Privacy preserving edge-cloud video analytics



FEVA: FEderated Video Analytics

- Assumption (threat model): 3D re-contruction does not contain privacy related information

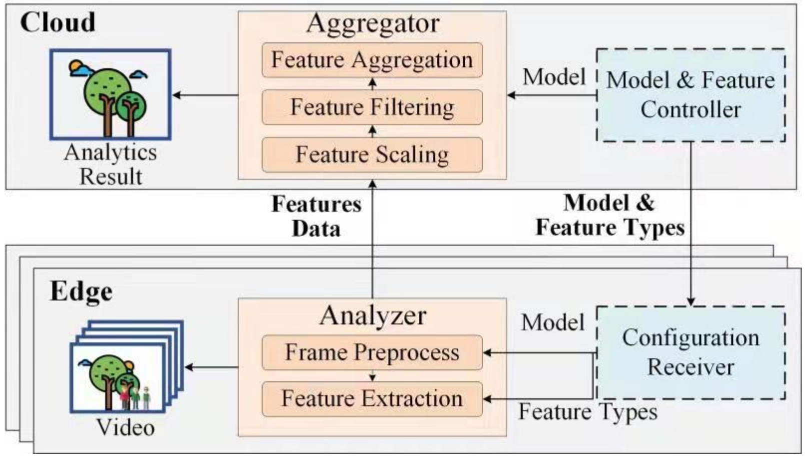
- Application-agnostic: build on top of TensorFlow Federated (TFF, Google's federated optimization framework)
- Observation: there are sensitive computation and insensitive computation

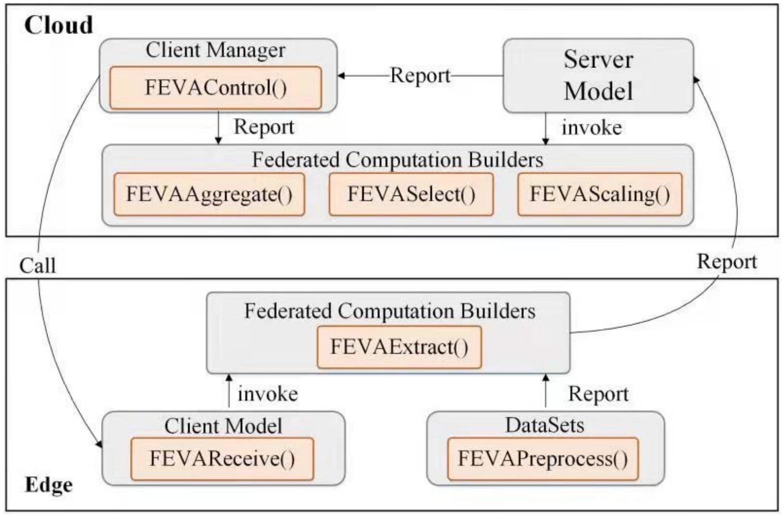


Implementation

- FEVA: A FEderated Video Analytics Architecture for Networked Smart Cameras

- Implementation on TFF, open source:
- https://github.com/polyuDLab/FEVA-DEMO





Problem and solution

- The FEVA Resource Optimization (FEVA-RO) Problem: Given the computation and the communication capacity of the edge devices and the cloud, the pre-trained NN models, the features, and the required delay, determine the NN model and the set of features for the video analytics task to maximize the analytic accuracy.

- There is an embedded Knapsack problem – NP-hard

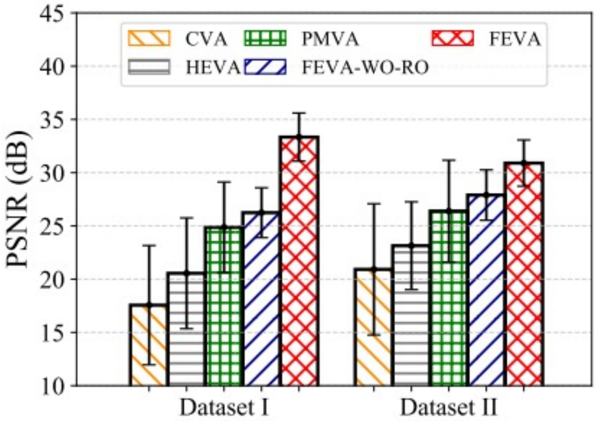
- Model selection algorithm – to determine the NN model
- Feature selection algorithm – to determine the features

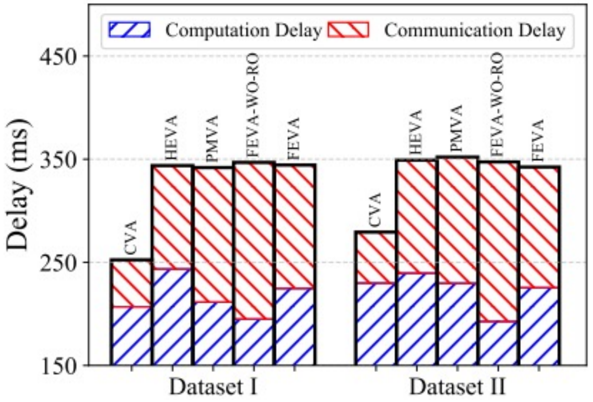
Evaluation setups

- Baselines:
  - Collaborative Video Analytics (CVA): the edges ignore the frames with private information.
  - Homomorphic Encryption Video Analytics (HEVA): the edges encrypt the data before uploading to the cloud server and the cloud conduct homomorphic analytics.
  - Privacy Masking Video Analytics (PMVA): the edges detect and replace private information with null values.
- Dataset: 1) VeRi, contains over 50,000 images of 776 vehicles and 2) CityFlow, contains more than 10,000 images of 666 vehicles.
- Metrics: 1) accuracy: PSNR, 2) delay

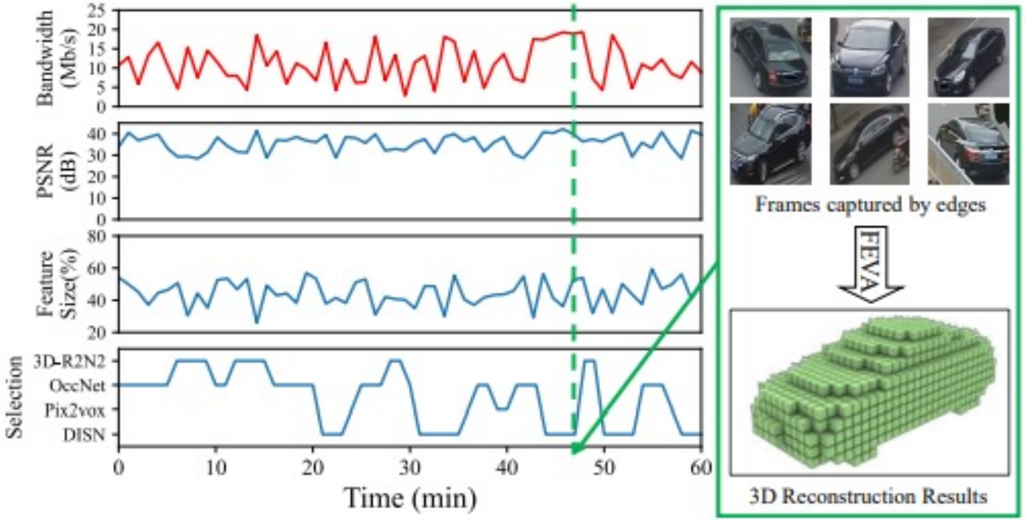
Evaluation

- Evaluation results





- An end-to-end operation of FEVA



Opportunities and challenges

Opportunities and Challenges

- Architecture, resource optimization, applications
  - Architecture: on top of existing TensorFlow Federated vs. dedicated stack
  - Resource optimization: computing, data, communications
  - Applications: HD maps, wireless networks, smart buildings

- Modeling, optimization, algorithms
  - Distributed computing, federated optimization
  - Data analytics task specific optimization
  - Stochastics, robustness, incentive designs

- Data mining, privacy, security
  - Model inversion attack
  - Local positioning attack

Opportunities and Challenges

- Federated analytics on privacy-preserving systems
  - E.g., Applying federated analytics to improve existing federated learning systems

- Federated analytics for privacy-demanding applications and systems
  - E.g., adding privacy elements to applications

Summary

- Federated Analytics, a new distributed computing paradigm
- What is FA in the research literature and Why FA from applications
- Federated video analytics: a first example
- Opportunities and Challenges

- Next to come: a few more examples on federated analytics

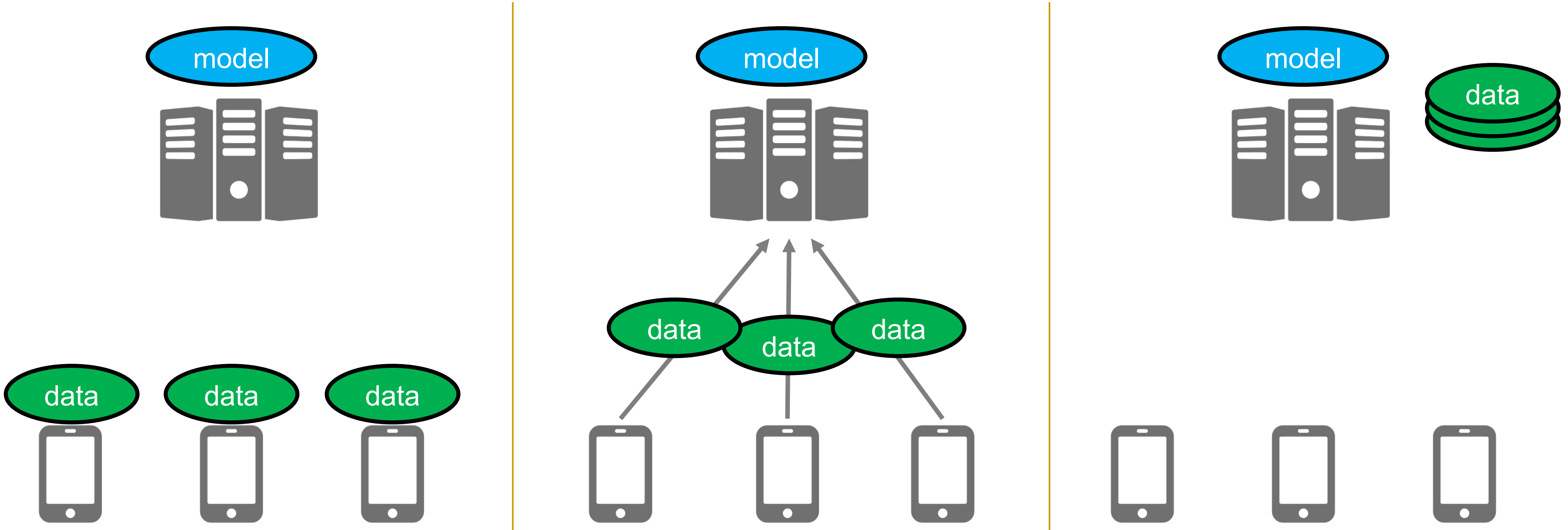
Examples on Federated Analytics

- Federated analytics for enhancing federated learning
  - FedACS : Federated Skewness Analytics in Heterogeneous Decentralized Data Environments
  - FAA-DL: Federated Anomaly Analytics for Local Model Poisoning Attack
- Federated analytics for privacy-demanding systems
  - FedFPM: A Unified Federated Analytics Framework for Collaborative Frequent Pattern Mining

How to Overcome Data Heterogeneity in Federated Optimization with Federated Analytics ?

Background

- Traditional DS/AI workflow: gather-and-analyze



① Central model & edge data

② Gather data

③ Central training

Z. Wang, Y. Zhu, D. Wang and Z. Han, “FedACS: Federated Skewness Analytics in Heterogeneous Decentralized Data Environments,” *IWQOS 2021*

Background & Motivation

- Increasing privacy awareness challenges the gather-and-analysis paradigm

> *Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject.*
>
> *—— EU General Data Protection Regulation (GDPR)*

- **Federated optimization**: an attempt to utilize edge data in the age of privacy protection

Federated Optimization

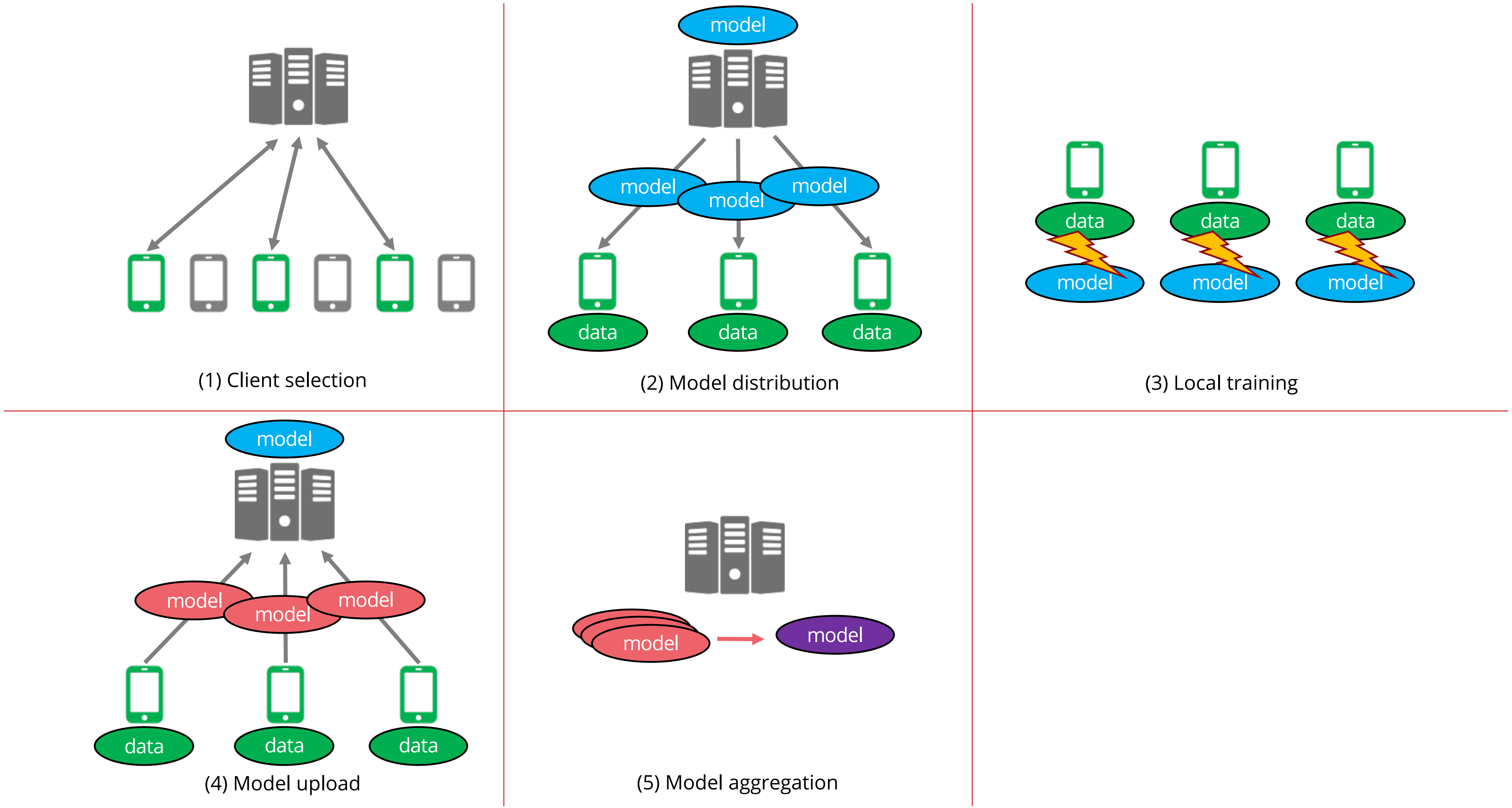
- In federated optimization:
  - Part of the AI/DS task is conducted by the clients
  - Model is transmitted between the server and clients (edge devices)
  - The server aggregates the results based on the models

- Benefits:
  - Server only knows **abstraction** of raw data
  - Utilize edge computation power
  - Reduce communication overhead

Federated Learning

- Federated learning (FL) is the earliest and most popular scenario of federated optimization
- It targets on **training a neural network**
- The training process is performed iteratively
- In each iteration:
  1. Select participated clients
  2. Distribute the neural network to the selected clients
  3. Gradient descent based on the local data
  4. Upload the updated neural network
  5. Aggregate the network with some aggregation rules

Federated Learning



(1) Client selection

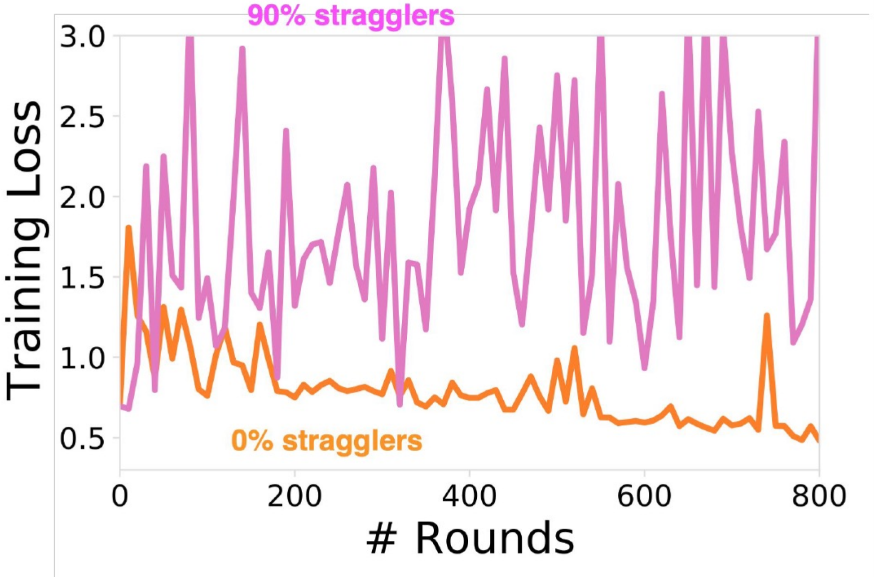
(2) Model distribution

(3) Local training

(4) Model upload

(5) Model aggregation

Device Heterogeneity

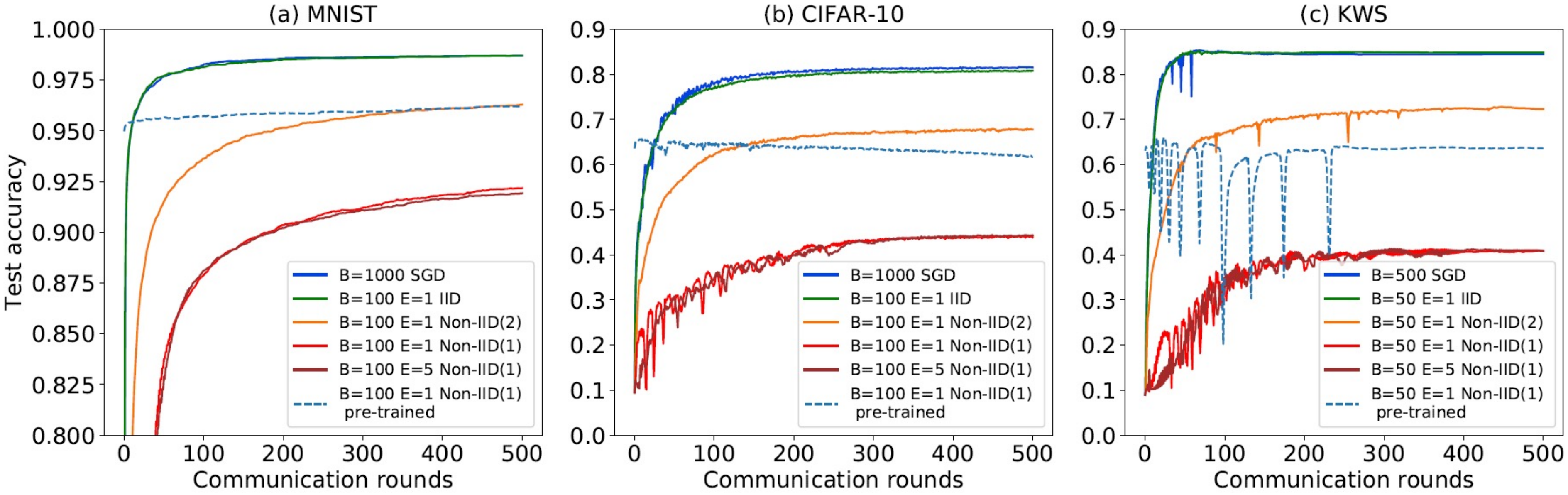


Device heterogeneity (e.g. clients that have limited resource and are likely to drop) hinders the convergence of federated optimization

[Li et al, Federated optimization in heterogeneous networks, MLSys 2020]
[Kairouz et al, Federated learning tutorial, NeurIPS 2020]

Data Heterogeneity

SGD: centralized training
IID: independent and identical distributed
Non-IID (1): each client owns data of 1 class
Non-IID (2): each client owns data of 2 classes

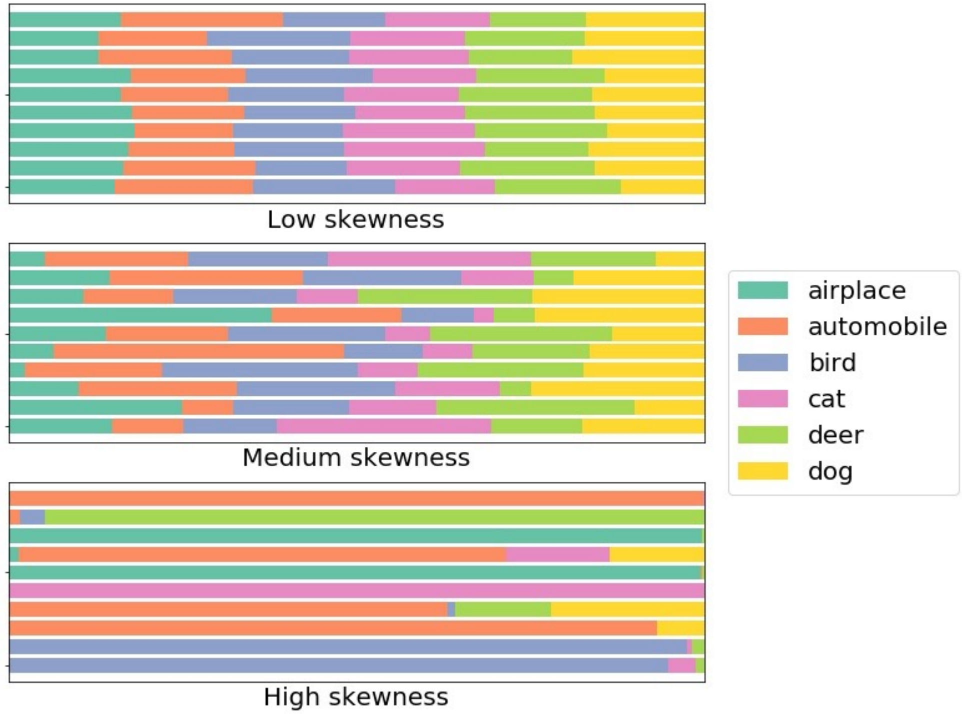


Data heterogeneity (Non-IID data partition) leads to lower FL accuracy and slower convergence

[Zhao et al, Federated learning with non-IID data, arxiv]

Diverged Data Heterogeneity

- Data heterogeneity: class distribution of the client data is skewed
- Skewness: the severity of data heterogeneity



Client data in federated optimization are usually skewed

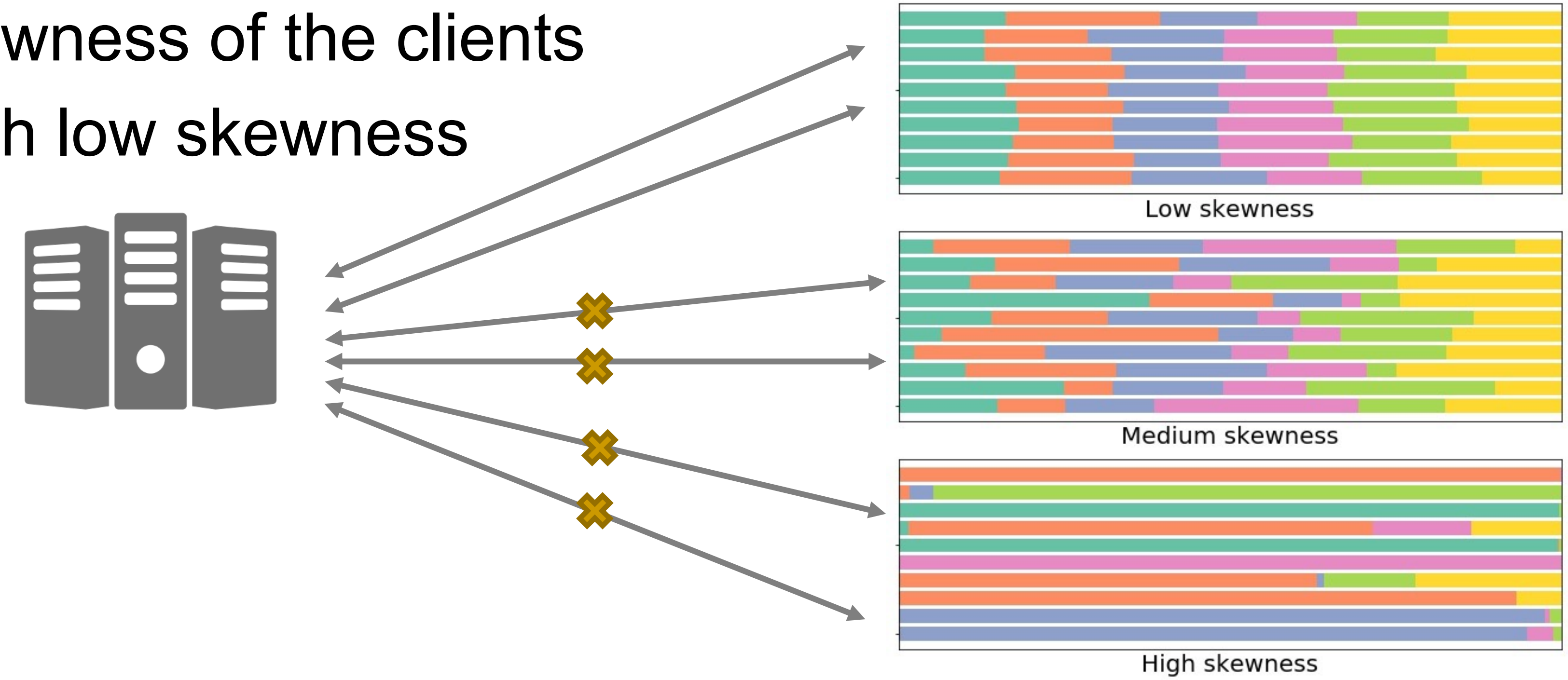
The exists clients with different skewness in the system

Clients with different skewness are not equally beneficial for the federated optimization system

Clients with low skewness provide more benefit

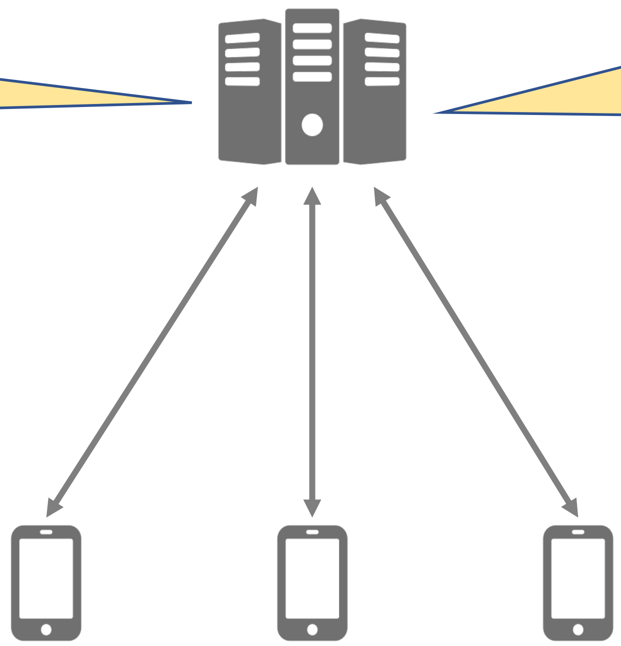
FedACS: Federated Skewness Analytics and Client Selection

- FedACS: an extra FA instance to assist other federated tasks
- Idea of FedACS
  - Measure the skewness of the clients
  - Select clients with low skewness



Heterogeneity-aware Client Selection

- Three steps



Step 2: Skewness estimation
Server aggregates the insights and infer about client skewness

Step 3: Client selection
Server selects the participating clients based on the skewness estimation

Step 1: Insight generation
Clients generate insights about the skewness of its local data

Heterogeneity-aware Client Selection

- Challenges

Step 1: Insight generation

- The insight should be informative about the client skewness
- The insight should be indirect to protect raw data privacy

Step 2: Skewness estimation

- It should derive useful knowledge from the indirect insights
- The procedure should be mathematically sound

Step 3: Client selection

- The selection should be robust to the system uncertainty
- The selection should satisfy requirements of the host tasks

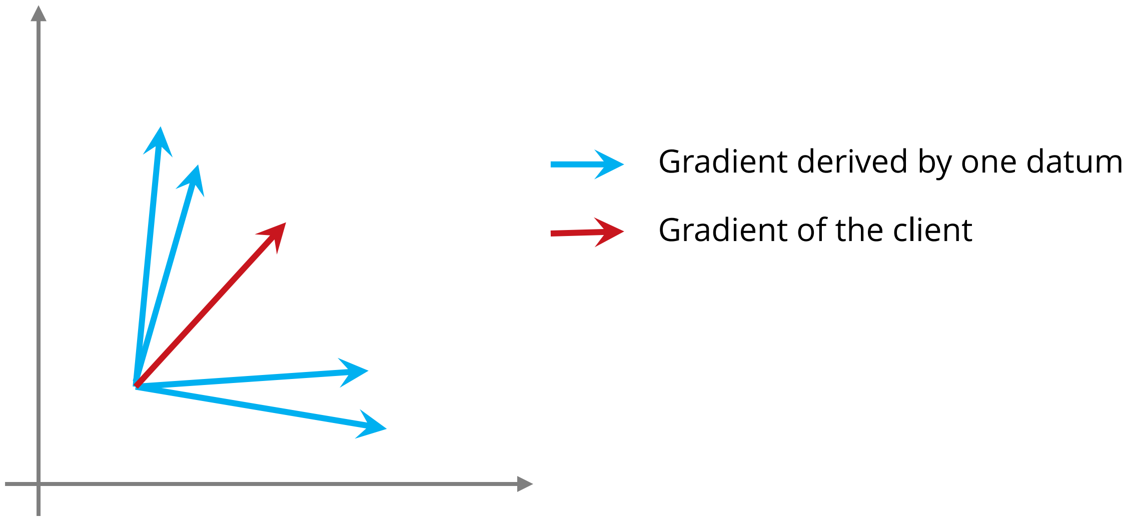
Step 1: Insight Generation

- The insight generation is formulated as gradient descent
  - Weight change of the neural network is used as insight
- Consistent to its host task, federated learning

- Benefits:
  - Do not need to install new computation scheme on the clients
  - Reuse the model distribution of FL, and reduce communication
  - Preserve the privacy protection level as FL

Step 2: Skewness Estimation

- Key idea: gradient (weight) from one client is the average of gradient derived by each individual data of the client



Step 2: Skewness Estimation

- Hoeffding's inequality
  - Provides possibility bound of average values diverging from their exception

Hoeffding's inequality: Supposed $X_1, \dots, X_n$ are independent variables, $X_i \in [a_i, b_i], \bar{X}$ is the average of X_i, there's

$$\Pr(|\bar{X} - E(\bar{X})| \geq \epsilon) < 2\exp(\frac{2\epsilon^2 n^2}{\sum_{i=1}^{n}(b_i - a_i)^2})$$

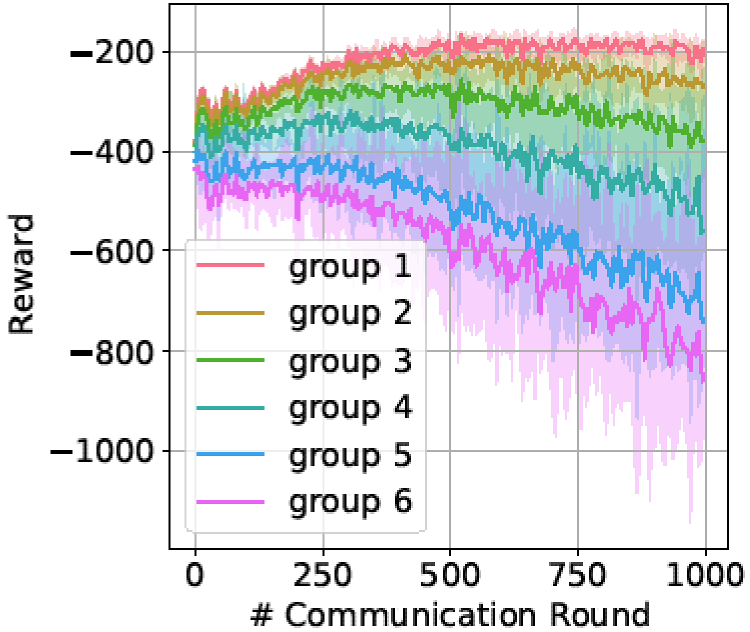
- Result of skewness estimation: higher R_i indicates lower skewness

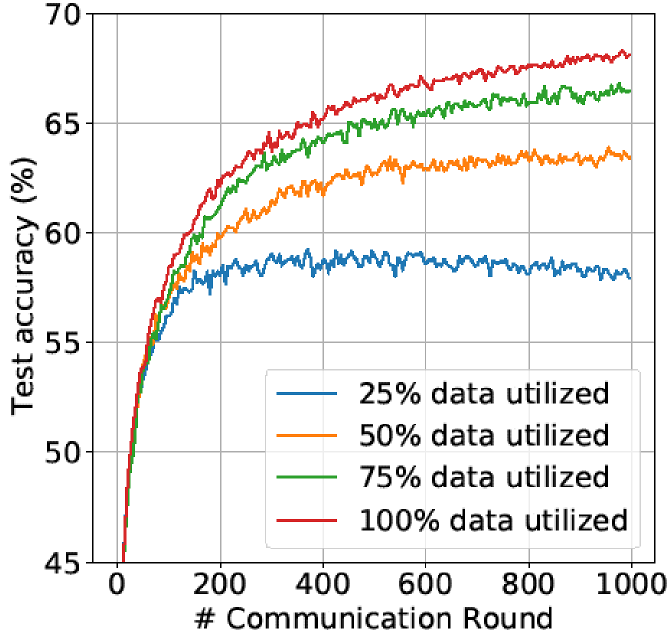
Denote Δw_i as the uploaded gradient from client i, and $\overline{\Delta w}$ as the average of uploaded gradients among all participating clients, there's

$$R_i = -\|\Delta w_i - \overline{\Delta w}\|_2$$

Step 3: Client Selection

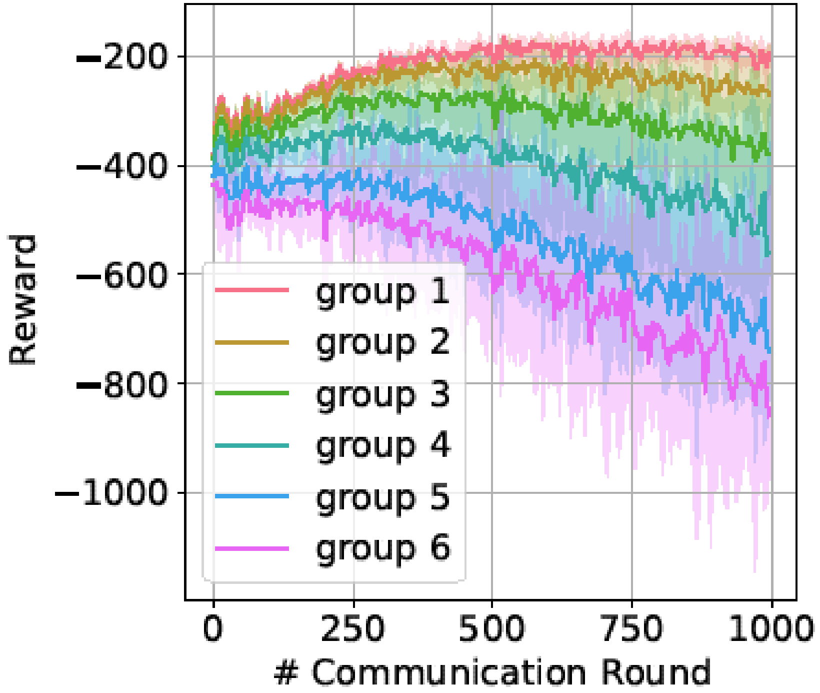
- Client selection is formulated as a multi-armed bandit
- Challenge #1: the R_i values is drifting in different rounds
- Challenge #2: the neural network needs sufficient training samples





Challenge #1: drifting R_i

- Challenge #1: the R_i values is drifting in different rounds
- Group 1 has the lowest skewness
- In the same round, clients with lower skewness earn higher reward
- No such guarantee in different round

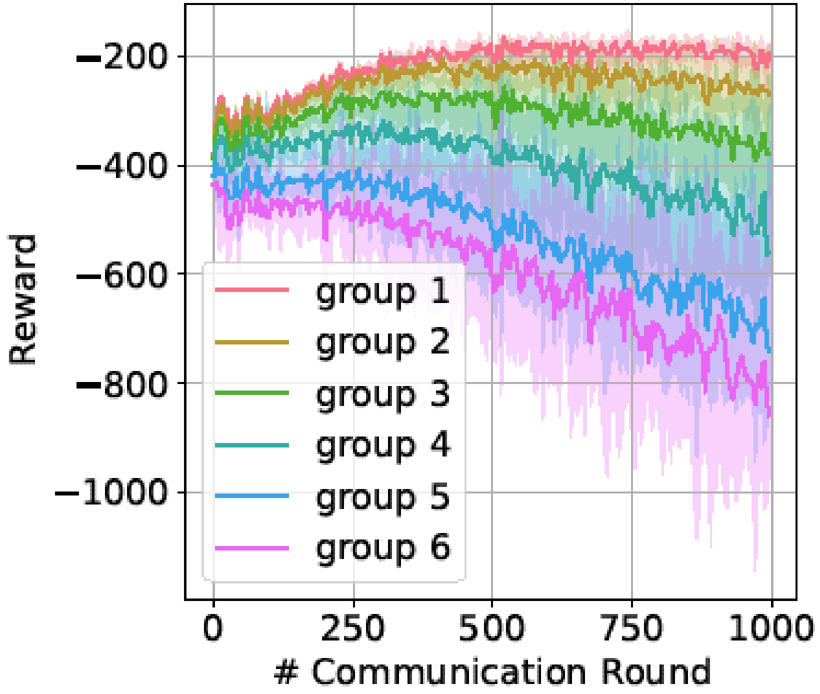


Challenge #1: drifting R_i

- Our solution: dueling bandit
  - Participating client “duel” with each other using their rewards
  - Train the bandit using the dueling results

$R_i = -2$ $\quad$ $R_j = -3$ $\quad$ $R_k = -10$

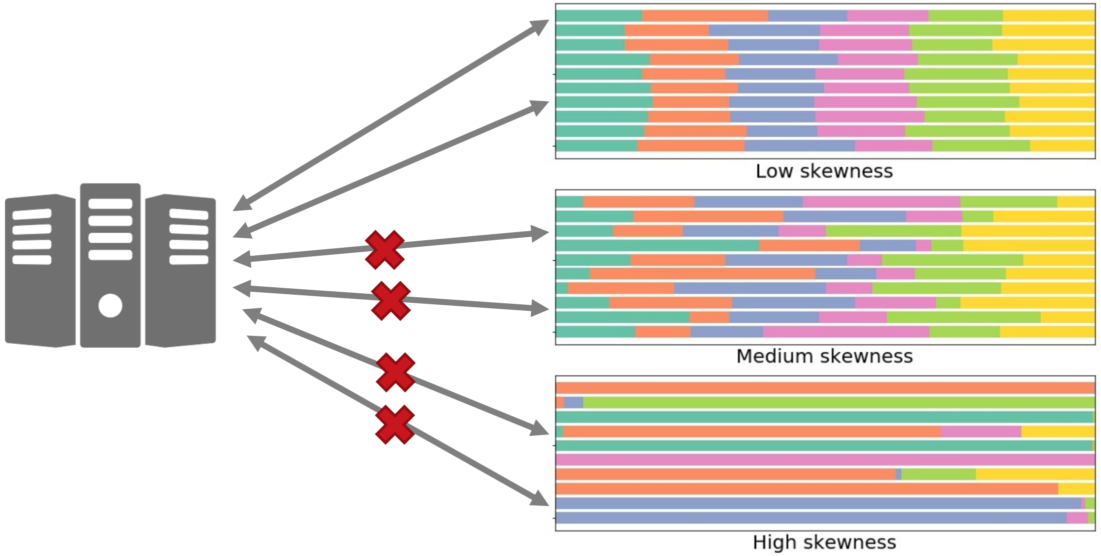
$win = 2, lose = 0$ $\quad$ $win = 1, lose = 1$ $\quad$ $win = 0, lose = 2$



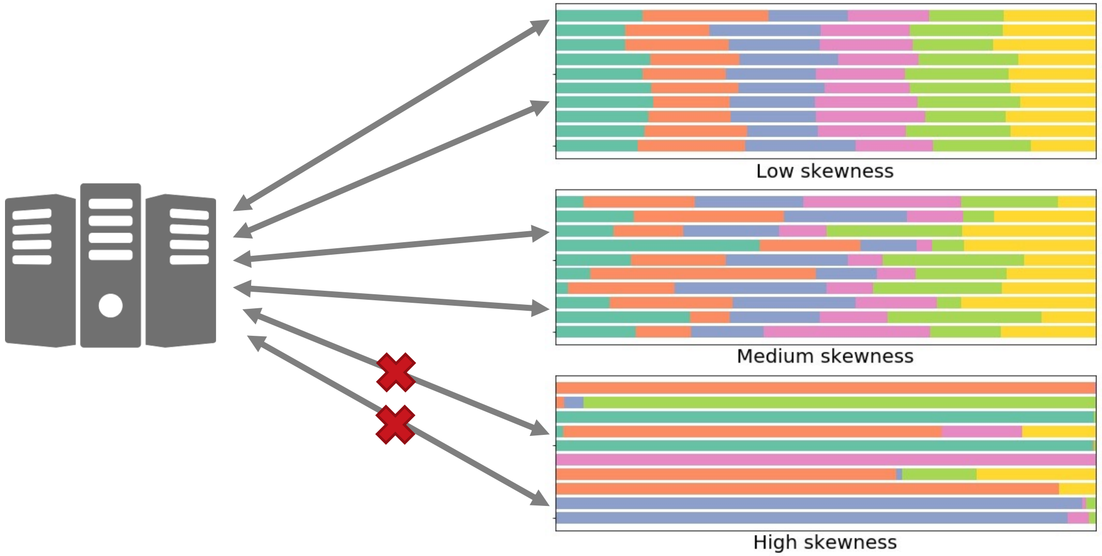
Challenge #2: restriction of training samples

- Challenge #2
  - The bandit targets at selecting the most perfect clients with low skewness
  - But the neural network will degrade due to lacking raw samples
- Our solution: set a parameter λ
  - There are N clients at all, and M participants in each round
  - The bandit find $\lambda \cdot N$ clients with low skewness to form a candidate pool
  - Then randomly draw M from the candidate pool as participants

Challenge #2: restriction of training samples



Low λ



High λ

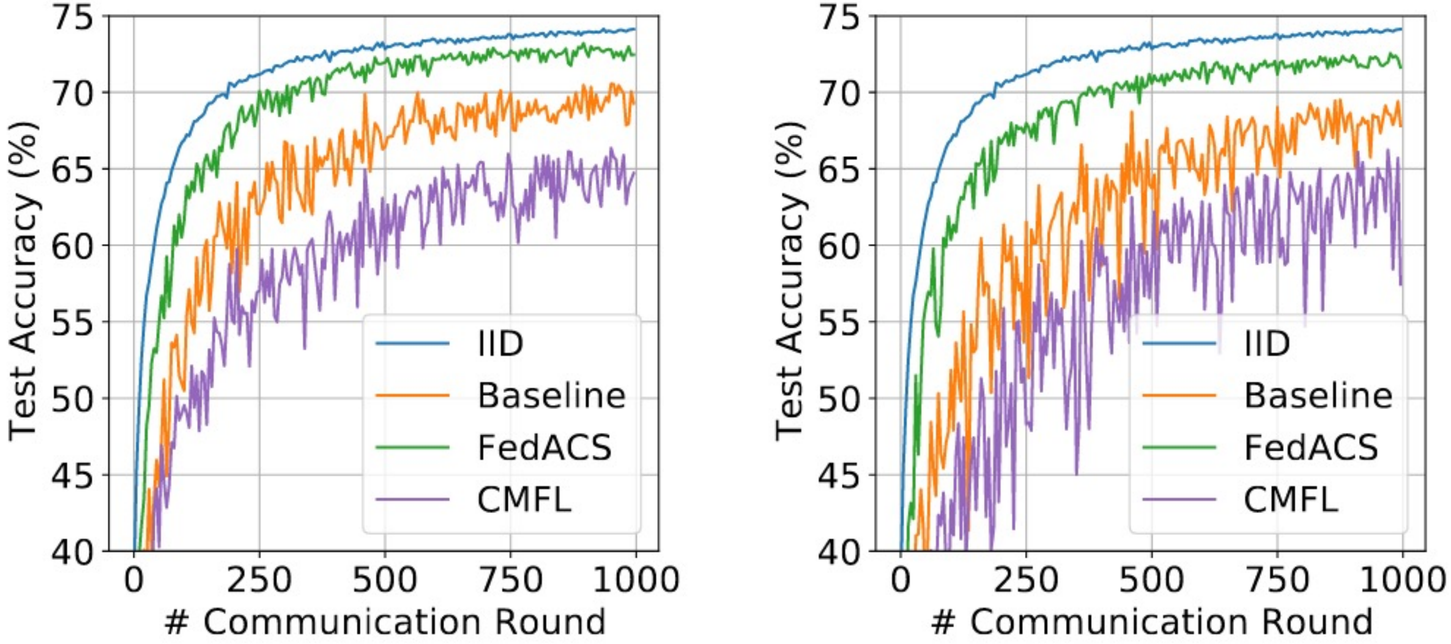
Evaluation: Setup

- Run FL on CIFAR10 dataset, using simple CNN in pytorch tutorial
- Two settings about data heterogeneity (low & high diversity)
- 10/200 clients selected in each round
- $\lambda = 0.4$
- Baseline: random selection
- Benchmark: CMFL[1]
  - Remove the “diverging” uploads based on the sign counts

[1] L. WANG, W. WANG and B. LI, "CMFL: Mitigating Communication Overhead for Federated Learning," *ICDCS* 2019.

Evaluation: Result

- FedACS restrains the degrading of FL caused by skewness

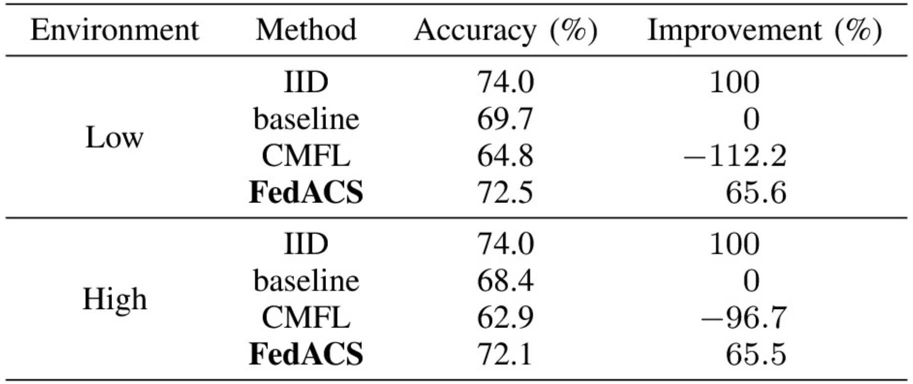


(a) Low

(b) High

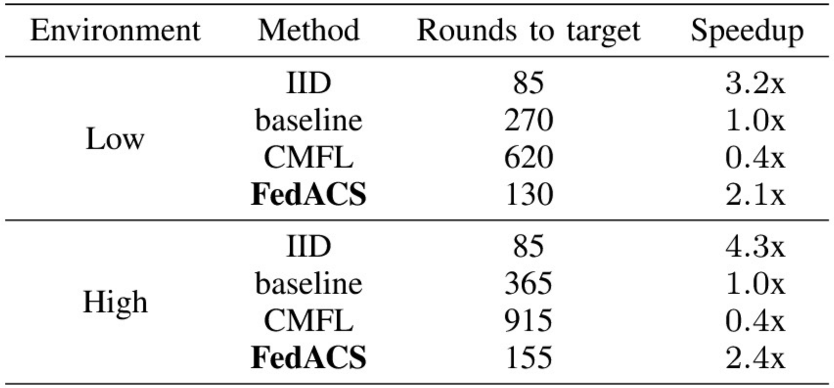
Evaluation: Result

- Details about accuracy



Environment	Method	Accuracy (%)	Improvement (%)
Low	IID	74.0	100
	baseline	69.7	0
	CMFL	64.8	−112.2
	FedACS	72.5	65.6
High	IID	74.0	100
	baseline	68.4	0
	CMFL	62.9	−96.7
	FedACS	72.1	65.5

- Details about convergence speed



Environment	Method	Rounds to target	Speedup
Low	IID	85	3.2x
	baseline	270	1.0x
	CMFL	620	0.4x
	FedACS	130	2.1x
High	IID	85	4.3x
	baseline	365	1.0x
	CMFL	915	0.4x
	FedACS	155	2.4x

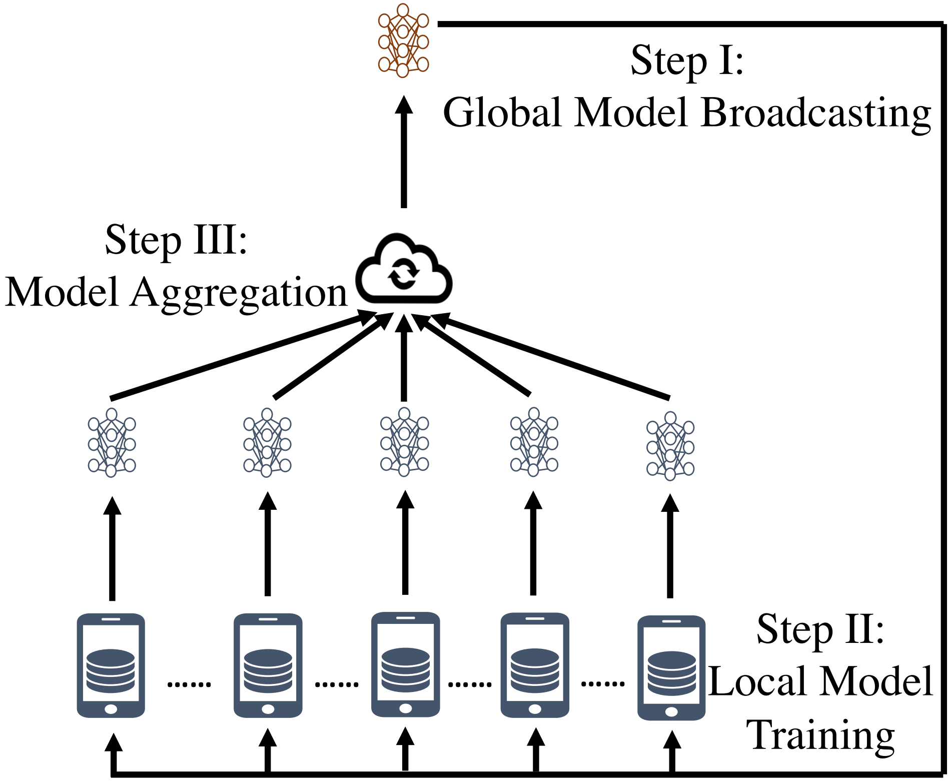
Summary

- FedACS: skewness estimation and client selection for federated optimization tasks
  - Small overhead insight derivation: reuse the infrastructure of the host task
  - Theoretically guaranteed skewness calculation: Hoeffding's Inequality
  - Robust client selection: dueling bandit and quality & quantity parameter
- Experiments show that FedACS significantly reduce the degrading effect caused by data heterogeneity

How to Improve the Robustness of Federated Learning with Federated Analytics ?

Background

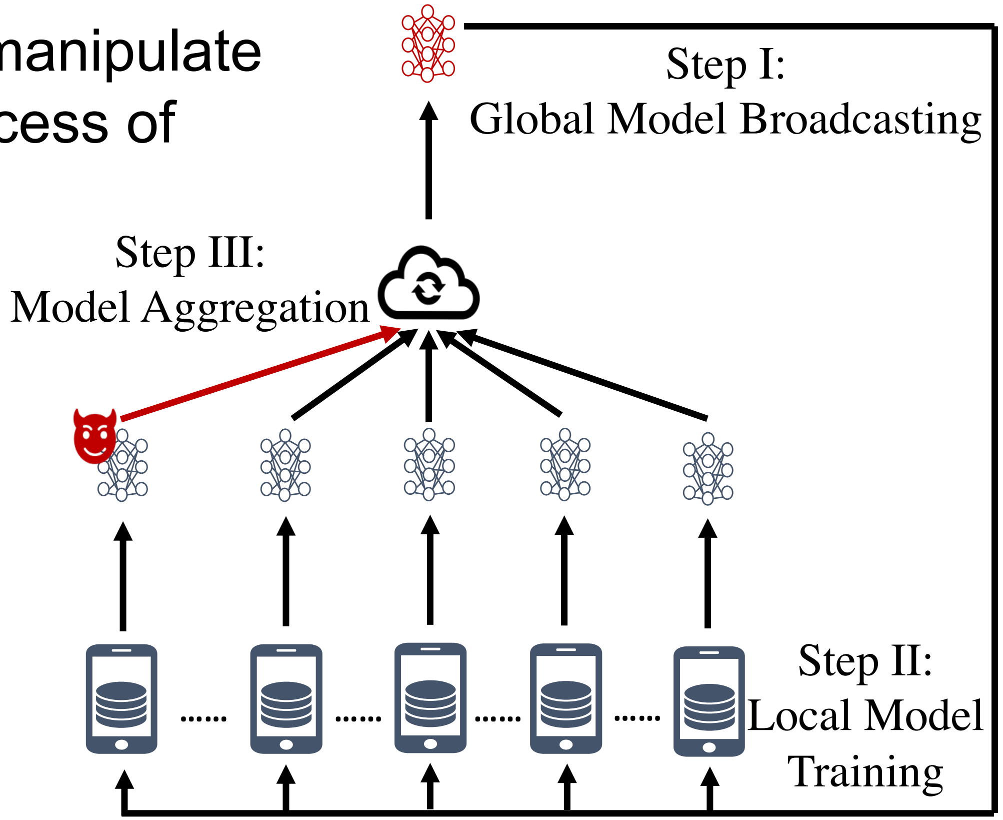
- Federated learning: Collaborative learning with decentralized data
  - Global model distribution
  - Model training with local data
  - Upload and aggregate local updates



S. Shi, C. Hu, D. Wang, Y. Zhu, Y. Zhu and Z. Han, “Federated Anomaly Analytics for Local Model Poisoning Attack,” JSAC 2021

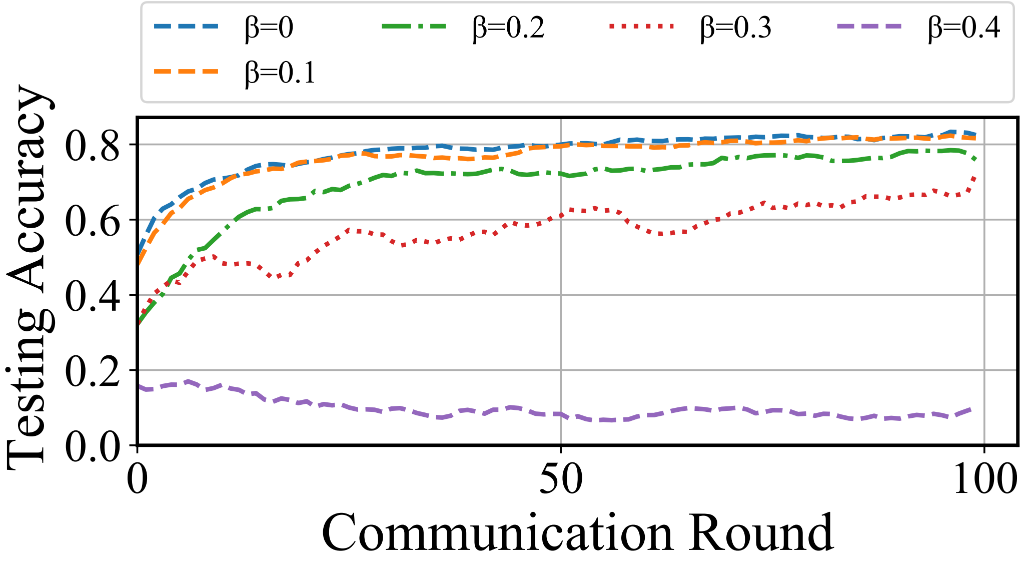
Background

- Federated learning is vulnerable to attacks
  - Local model poisoning attack
    - A single malicious worker can arbitrarily manipulate the uploaded local models during the process of federated learning



Background

- Federated learning is vulnerable to attacks
  - Harmful effect on the whole federated learning process
    - Broadly slowing down the convergence rate[1]
    - Significantly degrading the prediction accuracy of the learned global model[2]



[1] . Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” NeurPIS 2017

[2] . Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “Howto backdoor federated learning,” AISTATS 2020

Background

- Existing defense methods
  - Employing robust statistical methods to mitigate the negative impact of the poisoned local models
    - Trimmed mean[1]
    - GeoMed[2]
    - Krum[3]
  - Limitations
    - The negative impact of poisoned local models is just mitigated and not eliminated
    - Easily failed when encountered fine crafted local model poisoning attacks

Simple and decoupled

Passive defense

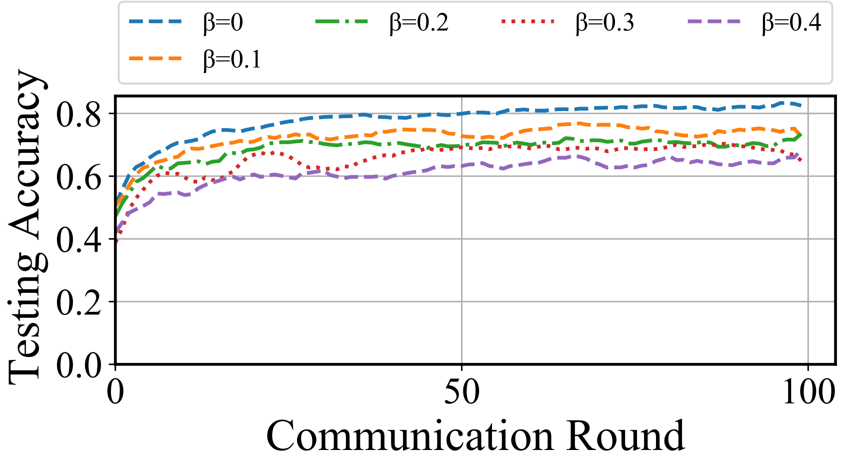
[1] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust dis-tributed learning: Towards optimal statistical rates,” ICML 2018

[2] . Chen, L. Su, and J. Xu, “Distributed statistical machine learningin adversarial settings: Byzantine gradient descent,” POMACS 2017

[3] . Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” NeurPIS 2017

Motivation

- Proactive defense
  - The learned model performance will improve when the number of malicious clients is decreased
  - Identify and remove the poisoned local models from aggregation



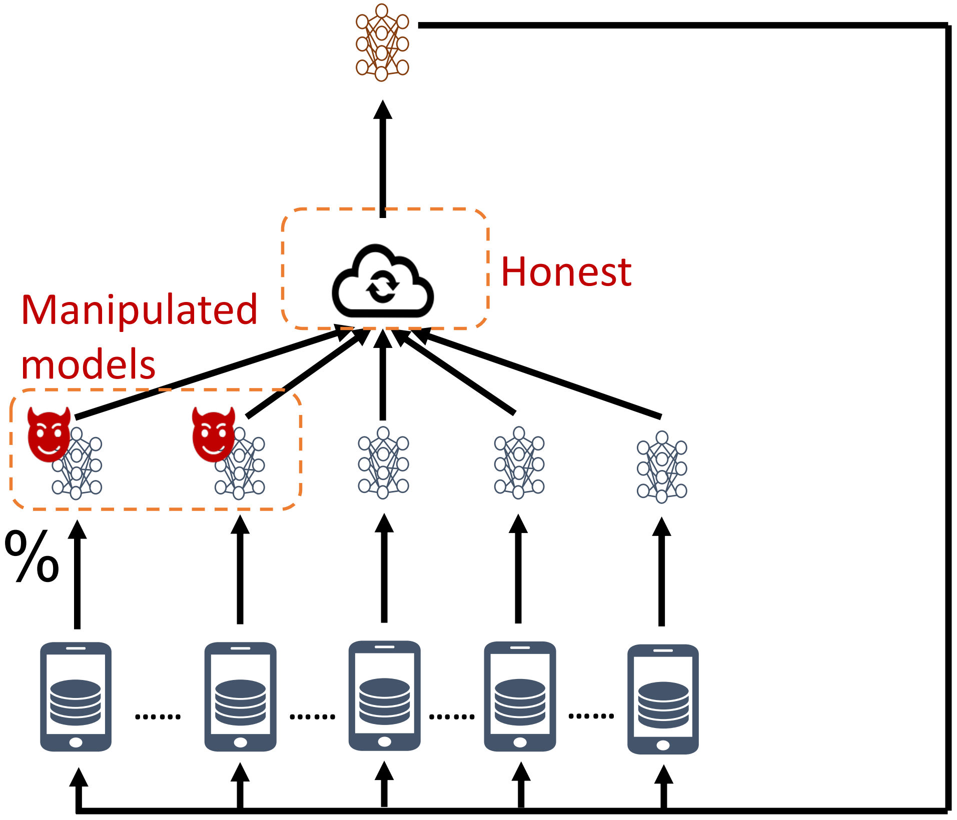
- Challenges
  - Identify the poisoned local models effectively
  - Protect the privacy of data in each local client

Motivation

- Our idea
  - Detect the uploaded local models with anomaly detection algorithm
  - Verify the poisoned local models with privacy guarantee by leverage federated analytics paradigm
  - Remove the verified poisoned local models from aggregation in each iteration

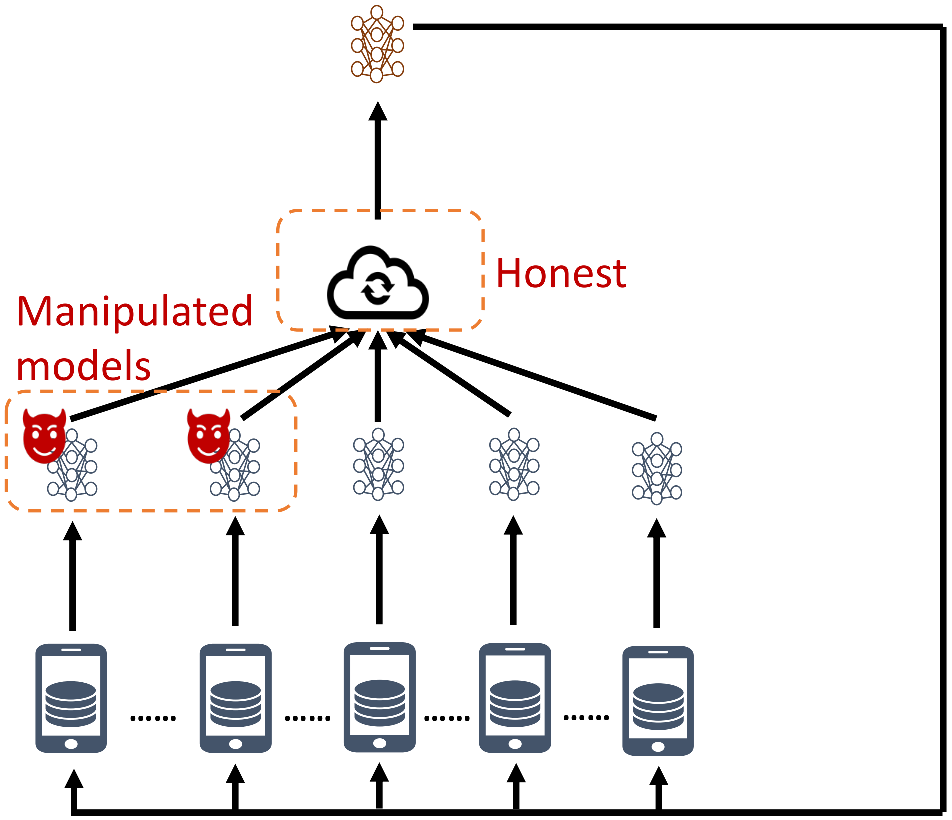
Federated Anomaly Analytics for Local Model Poisoning Attack

- Threat model
  - Server: curious but honest
  - Attacker: directly manipulates the local model updates (excluding the data poisoning attacks)
  - Proportion of the malicious clients: < 50%



Federated Anomaly Analytics for Local Model Poisoning Attack

- Capability of the attacker
  - Available to control the clients to be malicious but cannot control the server
  - The attacker has partial knowledge about the clients: it can only know the local model updates but not the local dataset

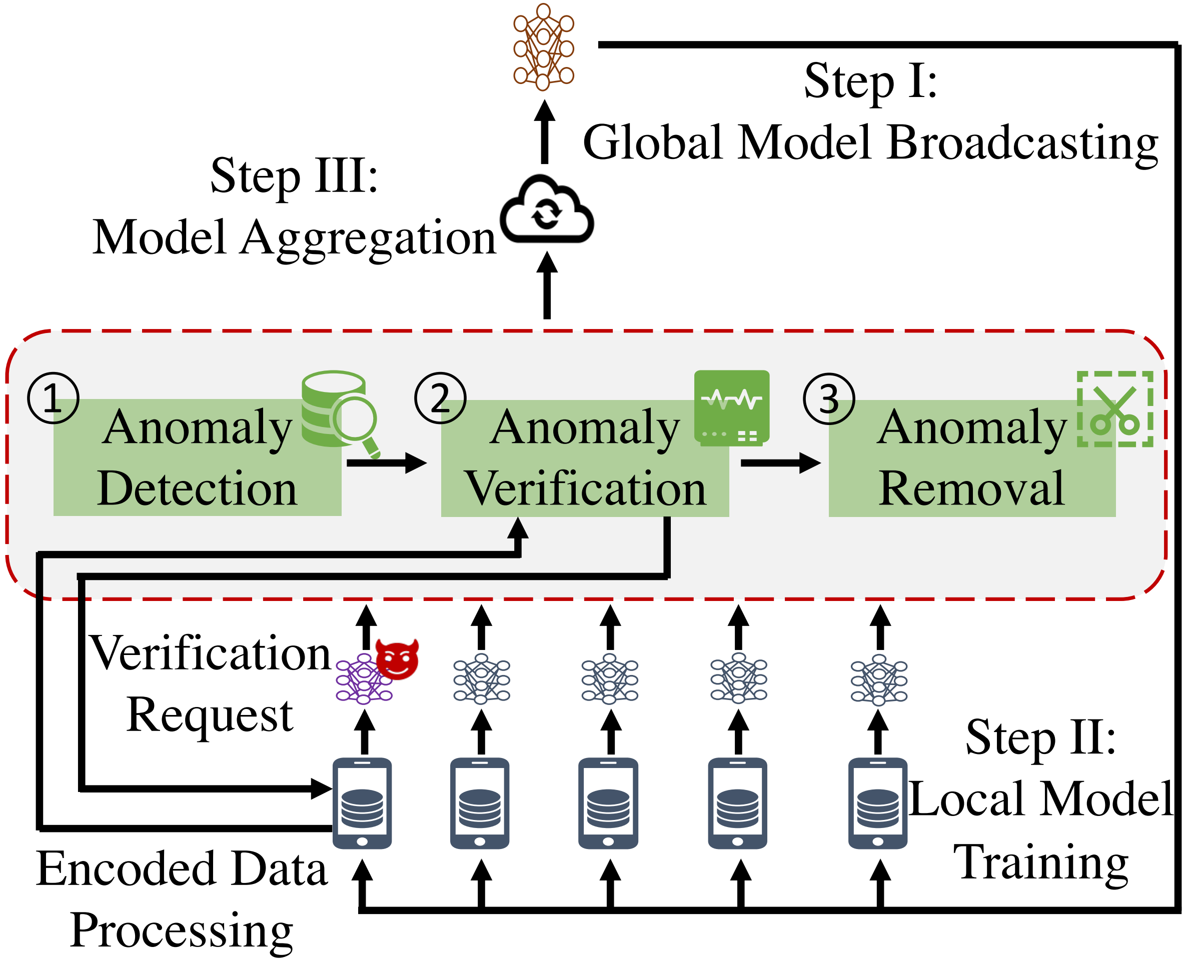


Federated Anomaly Analytics for Local Model Poisoning Attack

- Defense goals
  - **Fidelity**: The learned global model should be as accurate as the baseline learned when there is no attack
  - **Robustness**: The designed method should have the ability to defend against strong attacks (with a large proportion of malicious clients)
  - **Efficiency**: Much extra computation and communication overheads should not occur especially in resource-constrained edge devices.

Federated Anomaly Analytics for Local Model Poisoning Attack

- Framework overview
  - Anomaly Detection Module
  - Anomaly Verification Module
  - Anomaly Removal Module



Lightweight and Unsupervised Anomaly Detection

- Unsupervised Anomaly Detection algorithm
  - Filter out potentially malicious local model updates
  - Supervised anomaly detection methods have high accuracy but time-consuming and lacking labeled data in practice
  - We apply One-class SVM to identify poisoned and benign local models
  - Our framework allows greater compatibility with various anomaly detection algorithms

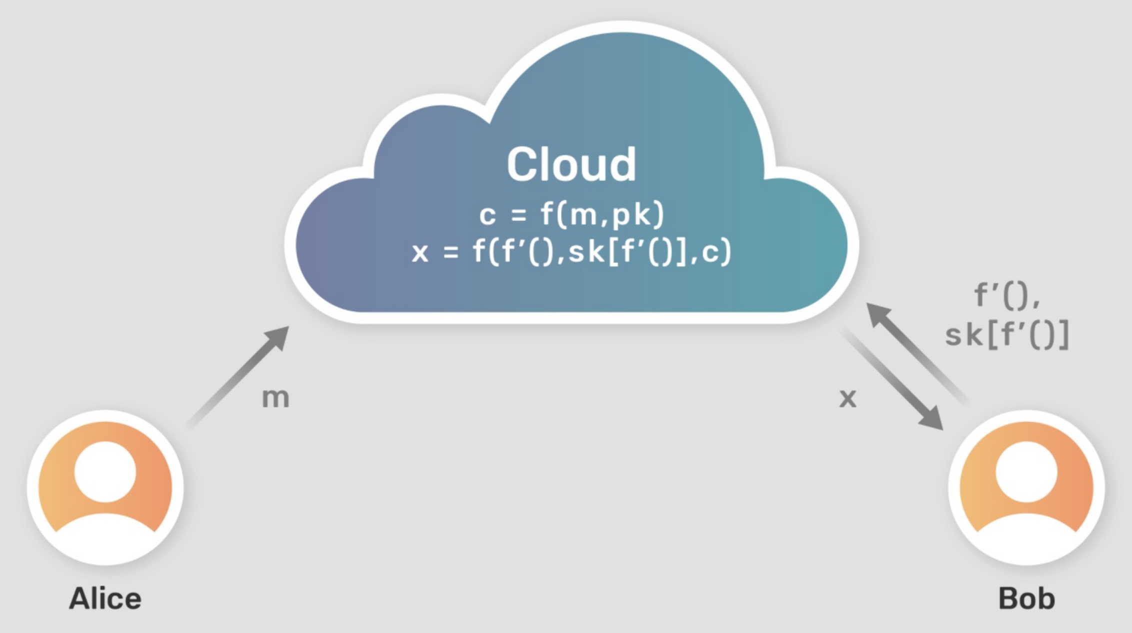
Privacy-preserving Anomaly Verification

- Verify the filtered potential malicious local models
  - Benign local models are misclassified to be potential malicious due to large variance of local models caused by Non-IID
  - Benign local models are trained with local data while the poisoned ones are not
  - Verify the truly poisoned local models by comparing them with the models trained by data in each local client
  - Each potential malicious client is requested to upload their local data in encoded format with privacy guarantee.

Privacy-preserving Anomaly Verification

- Privacy preserving for the uploaded local data
  - Functional Encryption
    - It enables to learn a function of what the ciphertext is encrypting, without revealing the plaintext.

$$f(x) \leftarrow Dec\left(mpk, sk_f, Enc(x)\right)$$



Privacy-preserving Anomaly Verification

- Privacy preserving for the uploaded local data
  - Inner product functional encryption (IPFE) scheme
    - It supports inner product function which is the common computation operation in model training
    - Only forward and backward propagation on the first layer of neural network model related to the training data

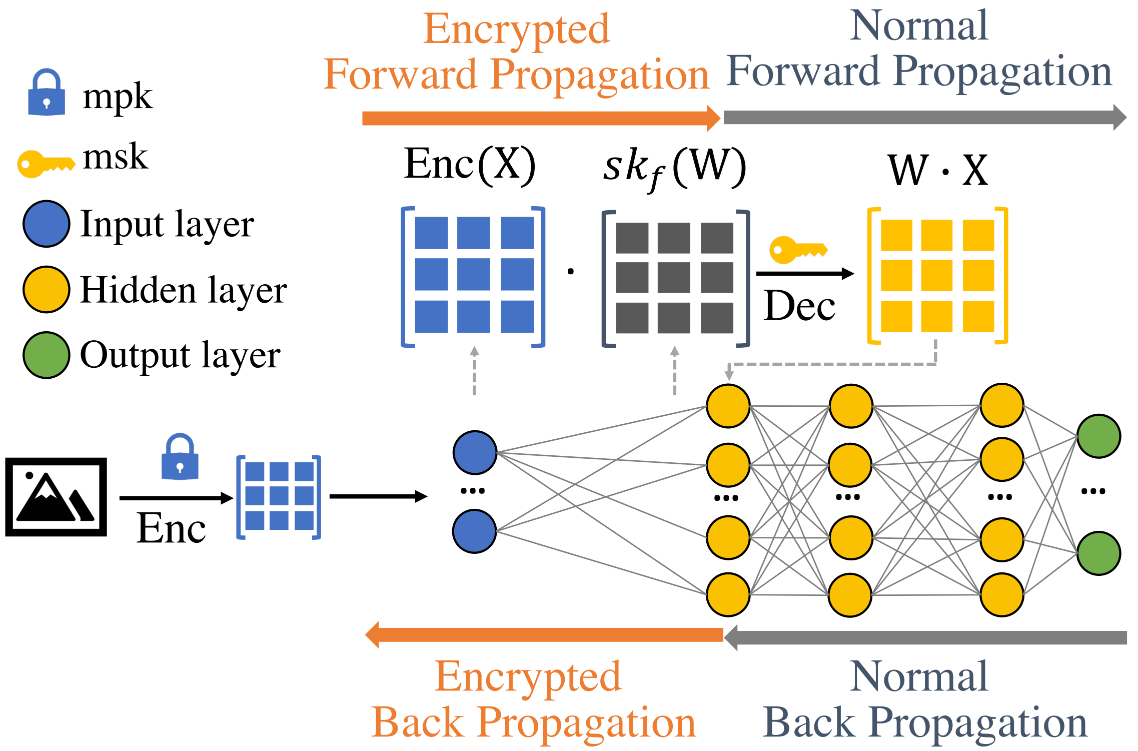
$$\mathbf{A}^j = \theta^j\left(\mathbf{Z}^j\right),$$

$$\mathbf{Z}^j = \mathbf{W}^j \cdot \mathbf{A}^{j-1} + \mathbf{b}^j,$$

$$\begin{aligned}\nabla \mathbf{W}^j = \frac{\partial E}{\partial \mathbf{W}^j} &= \frac{\partial E}{\partial \mathbf{A}^j} \cdot \frac{\partial \mathbf{A}^j}{\partial \theta^j} \cdot \frac{\partial \theta^j}{\partial \mathbf{Z}^j} \cdot \frac{\partial \mathbf{Z}^j}{\partial \mathbf{W}^j} \\ &= \frac{\partial E}{\partial \mathbf{A}^j} \cdot \frac{\partial \mathbf{A}^j}{\partial \theta^j} \cdot \frac{\partial \theta^j}{\partial \mathbf{Z}^j} \cdot \mathbf{A}^{j-1}.\end{aligned}$$

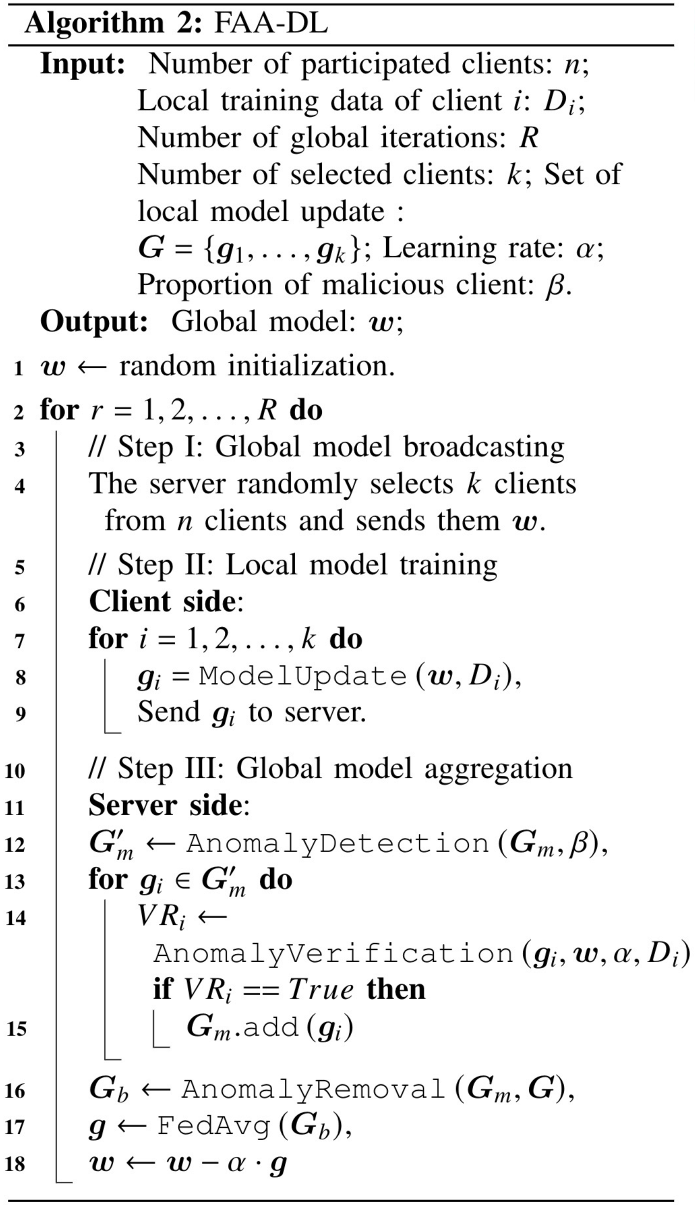
Privacy-preserving Anomaly Verification

- Privacy preserving for the uploaded local data
  - Training on encrypted data with IPFE
    - Obtain the decrypted local model
    - Compare with the uploaded local model
  - Verify the truly poisoned ones



Anomaly Removal

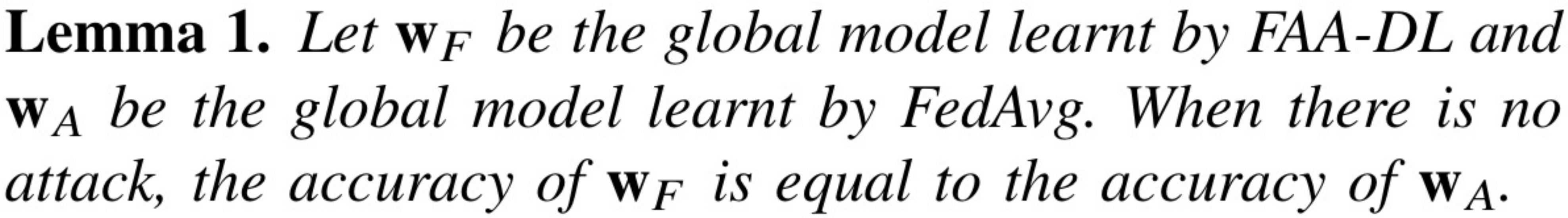
- Remove the verified poisoned local models from aggregation
- Update the global model with the aggregated local models



```
Algorithm 2: FAA-DL
Input:  Number of participated clients: $n$;
        Local training data of client $i$: $D_i$;
        Number of global iterations: $R$
        Number of selected clients: $k$; Set of
        local model update :
        $\boldsymbol{G} = \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k\}$; Learning rate: $\alpha$;
        Proportion of malicious client: $\beta$.
Output: Global model: $\boldsymbol{w}$;
$\boldsymbol{w} \leftarrow$ random initialization.
for $r = 1, 2, \ldots, R$ do
    // Step I: Global model broadcasting
    The server randomly selects $k$ clients
      from $n$ clients and sends them $\boldsymbol{w}$.
    // Step II: Local model training
    Client side:
    for $i = 1, 2, \ldots, k$ do
        $\boldsymbol{g}_i$ = ModelUpdate($\boldsymbol{w}, D_i$),
        Send $\boldsymbol{g}_i$ to server.
    // Step III: Global model aggregation
    Server side:
    $\boldsymbol{G}'_m \leftarrow$ AnomalyDetection($\boldsymbol{G}_m, \beta$),
    for $\boldsymbol{g}_i \in \boldsymbol{G}'_m$ do
        $VR_i \leftarrow$
          AnomalyVerification($\boldsymbol{g}_i, \boldsymbol{w}, \alpha, D_i$)
          if $VR_i == True$ then
              $\boldsymbol{G}_m$.add($\boldsymbol{g}_i$)
    $\boldsymbol{G}_b \leftarrow$ AnomalyRemoval($\boldsymbol{G}_m, \boldsymbol{G}$),
    $\boldsymbol{g} \leftarrow$ FedAvg($\boldsymbol{G}_b$),
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \cdot \boldsymbol{g}$
```

Theoretical Analysis

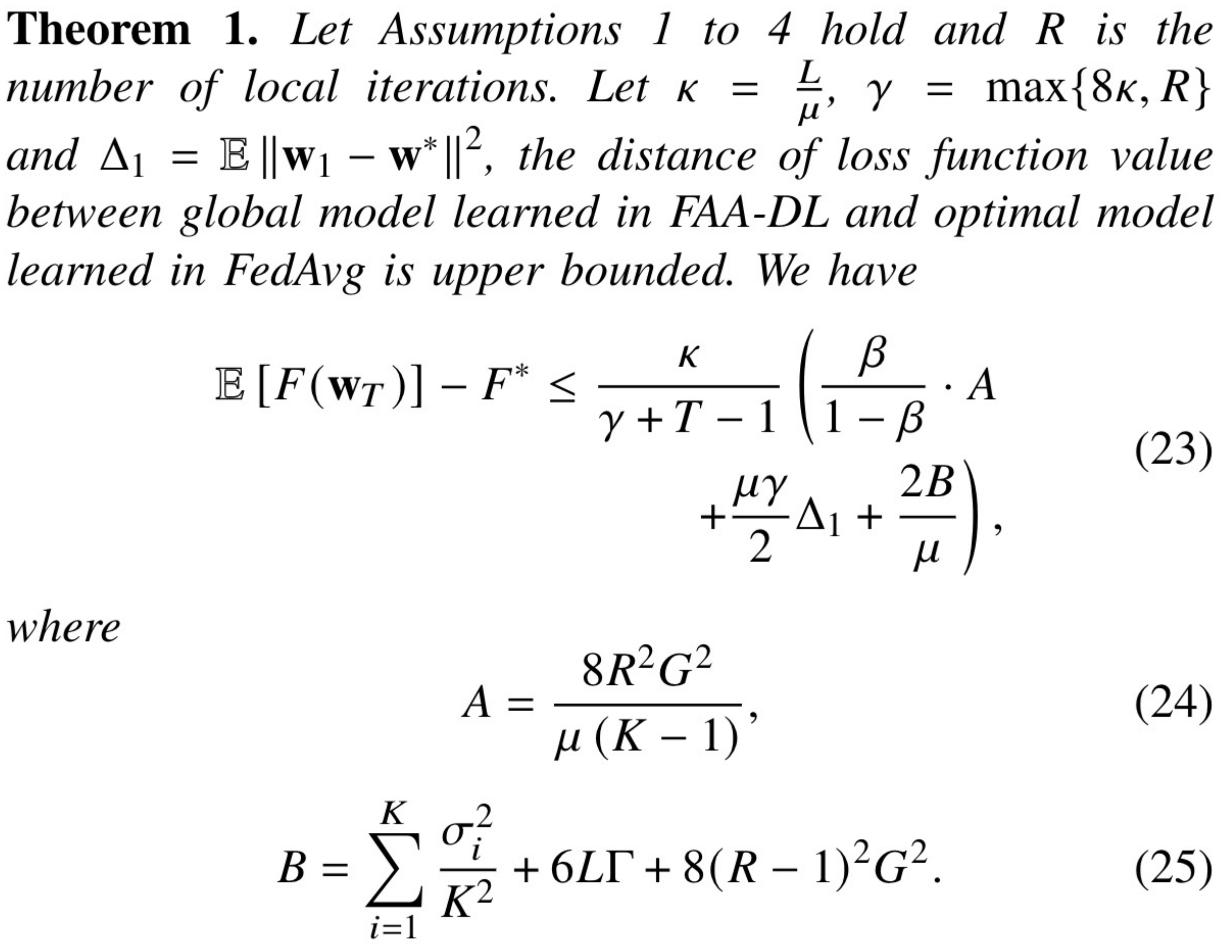
- Fidelity Analysis



Lemma 1. *Let* $\mathbf{w}_F$ *be the global model learnt by FAA-DL and* $\mathbf{w}_A$ *be the global model learnt by FedAvg. When there is no attack, the accuracy of* $\mathbf{w}_F$ *is equal to the accuracy of* $\mathbf{w}_A$.

Theoretical Analysis

- Robustness Analysis
  - The upper bound will always approximate to 0 even the value of β is largely increased, with $T \rightarrow \infty$



Theorem 1. *Let Assumptions 1 to 4 hold and R is the number of local iterations. Let* $\kappa = \frac{L}{\mu}$, $\gamma = \max\{8\kappa, R\}$ *and* $\Delta_1 = \mathbb{E}\left\|\mathbf{w}_1 - \mathbf{w}^*\right\|^2$, *the distance of loss function value between global model learned in FAA-DL and optimal model learned in FedAvg is upper bounded. We have*

$$\mathbb{E}\left[F\left(\mathbf{w}_T\right)\right] - F^* \leq \frac{\kappa}{\gamma + T - 1}\left(\frac{\beta}{1-\beta} \cdot A + \frac{\mu\gamma}{2}\Delta_1 + \frac{2B}{\mu}\right), \tag{23}$$

where

$$A = \frac{8R^2G^2}{\mu\left(K-1\right)}, \tag{24}$$

$$B = \sum_{i=1}^{K} \frac{\sigma_i^2}{K^2} + 6L\Gamma + 8(R-1)^2G^2. \tag{25}$$

Theoretical Analysis

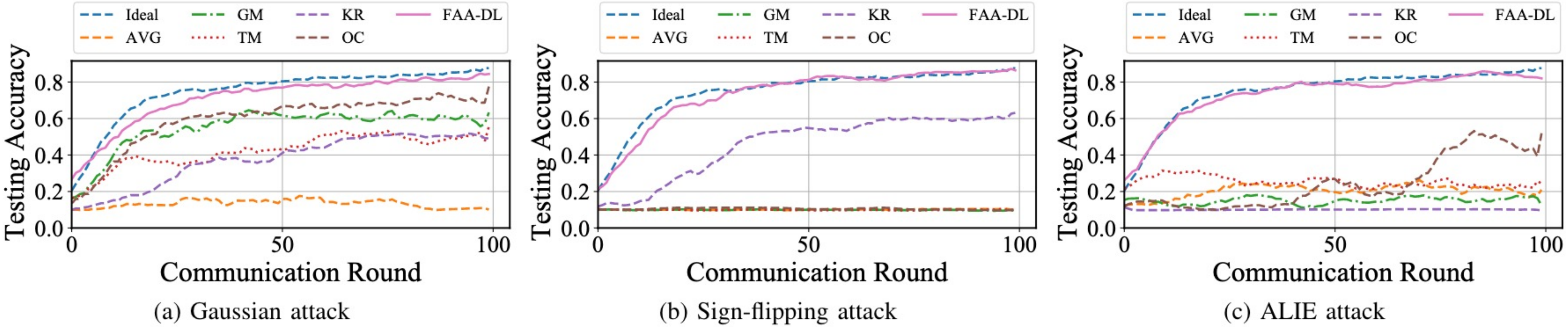
- Efficiency Analysis
  - Time complexity
    - Client : only a small amount of clients required to do encryption ($\mathcal{O}(1)$)
    - Server : Extra anomaly detection computation ($\mathcal{O}(n^3)$), negligible time consuming as the powerful computation capability of server

Evaluation

- Setup
  - Training dataset and model
    - MNIST and Fashion-MNIST with CNN
  - Attack model
    - Gaussian noise attack
    - Sign-flipping attack
    - ALIE attack
  - Defense benchmark
    - FedAVG (baseline)
    - Trimmed Mean
    - GeoMed
    - Krum

Evaluation

- Results
  - Accuracy
    - FAA-DL outperforms other defense methods on the accuracy of the learned global model, with an accuracy improvement up to 2.03X
    - The performance gap of FAA-DL is within 0.92% –2.48% of the ideal baseline across all tested attacks

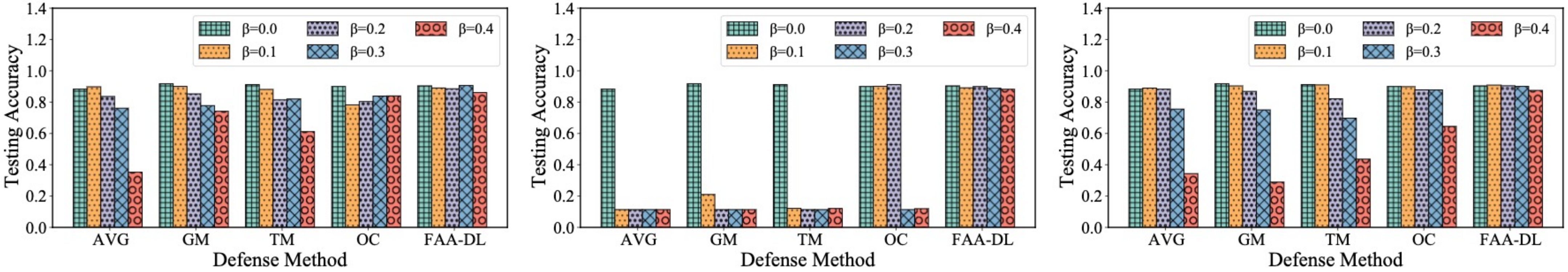


(a) Gaussian attack (b) Sign-flipping attack (c) ALIE attack

Fig. 5: The accuracy of defense to different attacks with different methods.

Evaluation

- Results
  - Robustness
    - FAA-DL remains nearly the same accuracy as the ideal baseline when the proportion of attacked devices increased from 0.1 to 0.4
    - Other defense methods decreased greatly especially in sign-flipping attack



(a) Gaussian attack

(b) Sign-flipping attack

(c) ALIE attack

Fig. 8: Top-1 accuracy of different defense methods to different attacks with various fraction of malicious devices(from 0.1 to 0.4)

Evaluation

- Results
  - Efficiency
    - The time cost of FAA-DL is approximate to the baseline and smaller than the benchmark trimmed mean

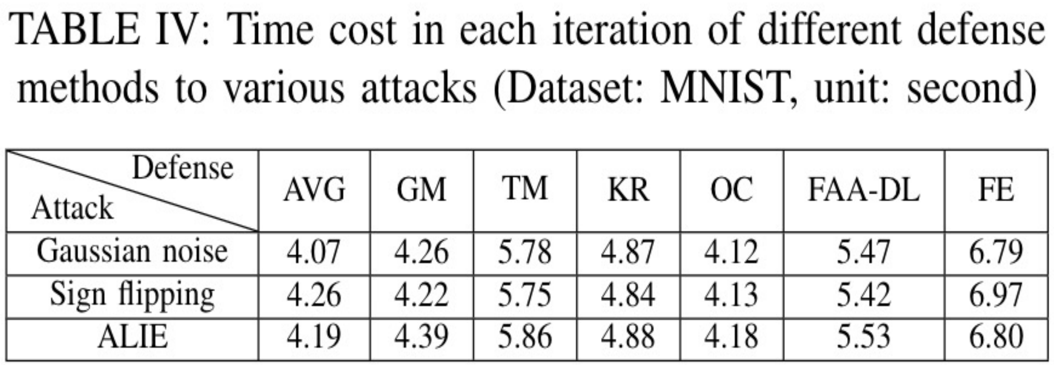


TABLE IV: Time cost in each iteration of different defense methods to various attacks (Dataset: MNIST, unit: second)

Defense Attack	AVG	GM	TM	KR	OC	FAA-DL	FE
Gaussian noise	4.07	4.26	5.78	4.87	4.12	5.47	6.79
Sign flipping	4.26	4.22	5.75	4.84	4.13	5.42	6.97
ALIE	4.19	4.39	5.86	4.88	4.18	5.53	6.80

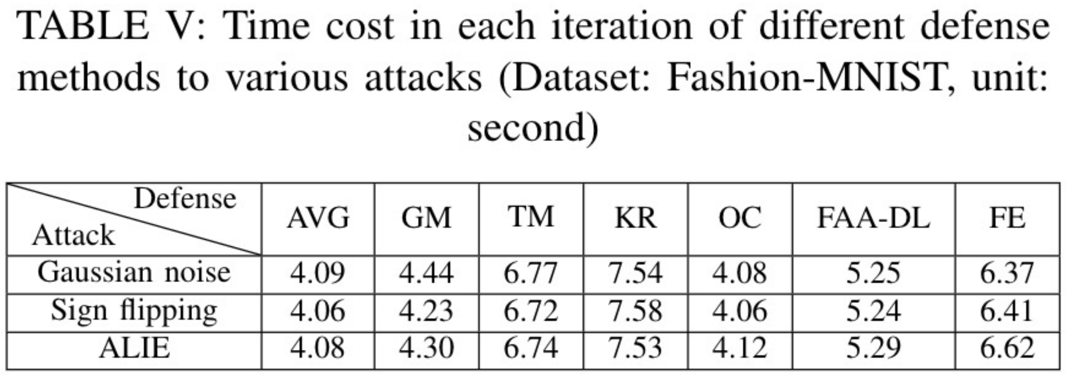


TABLE V: Time cost in each iteration of different defense methods to various attacks (Dataset: Fashion-MNIST, unit: second)

Defense Attack	AVG	GM	TM	KR	OC	FAA-DL	FE
Gaussian noise	4.09	4.44	6.77	7.54	4.08	5.25	6.37
Sign flipping	4.06	4.23	6.72	7.58	4.06	5.24	6.41
ALIE	4.08	4.30	6.74	7.53	4.12	5.29	6.62

Summary

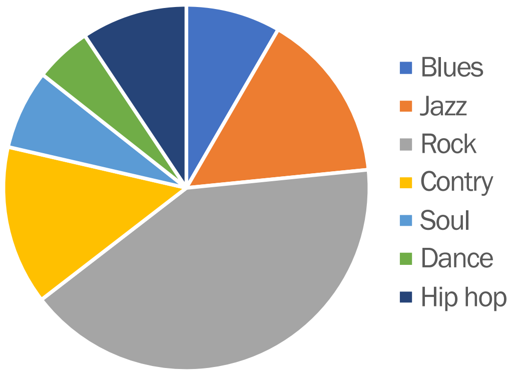
- FAA-DL: proactive defense with privacy guarantee
  - Light-weight anomaly detection: filter out potential poisoned local models
  - Privacy preserving anomaly verification: achieve by utilizing functional encryption method

- Experiment results show FAA-DL outperforms other defense methods with a robustness guarantee.

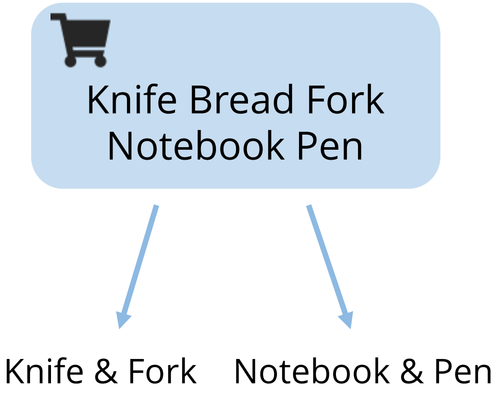
How to assist privacy-demanding data analytics with Federated Analytics ?

Background: frequent pattern mining

- Discover frequent patterns (items, subsets, subsequences) with frequency in population higher than the threshold



Frequent item mining

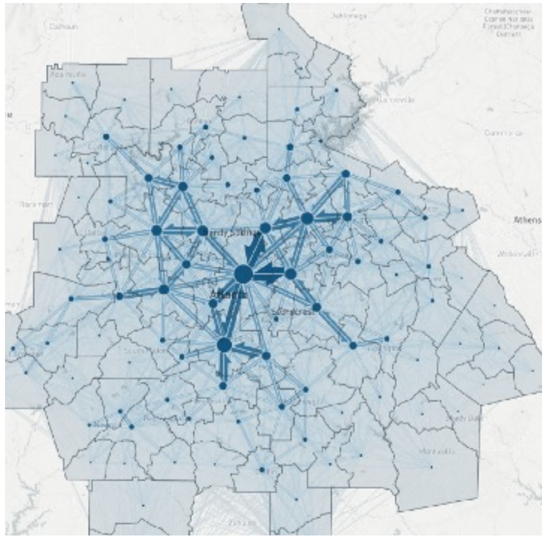


Knife Bread Fork Notebook Pen

Knife & Fork

Notebook & Pen

Frequent itemset mining



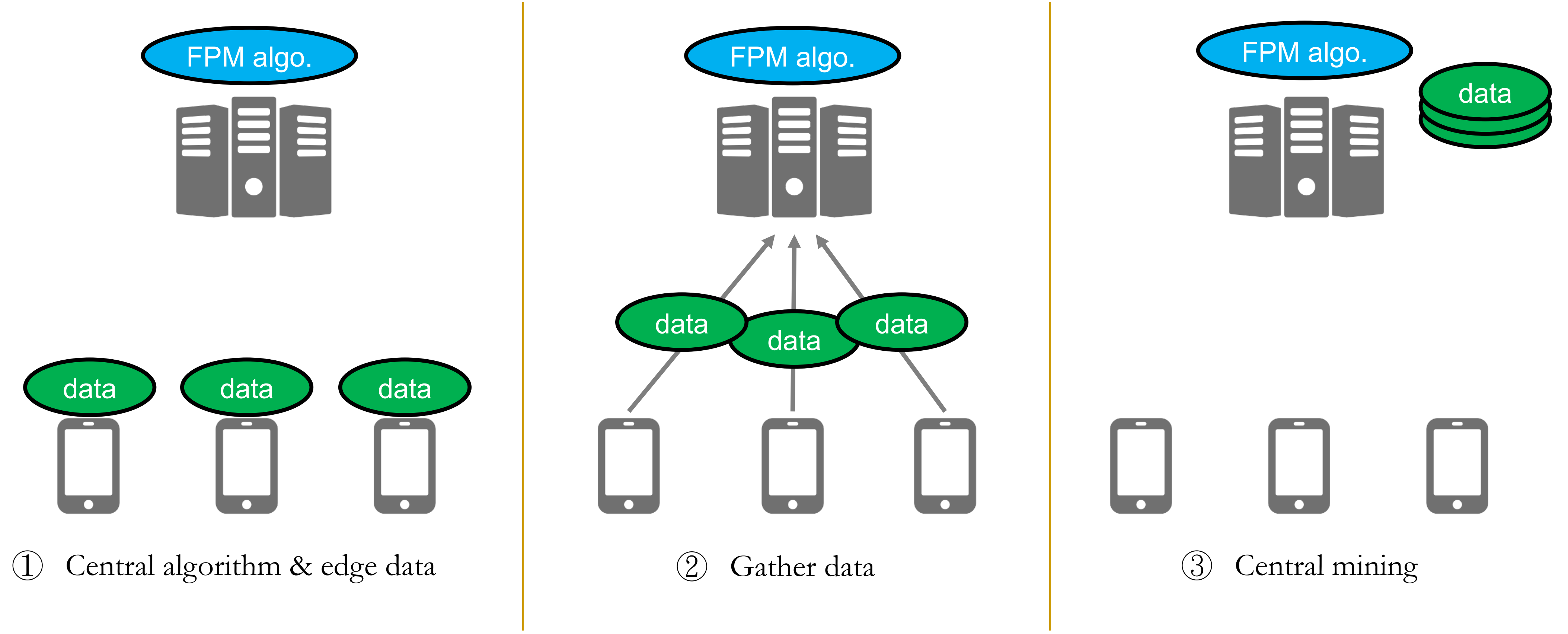
Frequent sequence mining*

[*Credit: http://www.teralytics.net/]

Z. Wang, Y. Zhu, D. Wang and Z. Han, “FedFPM: A Unified Federated Analytics Framework for Collaborative Frequent Pattern Mining”, INFOCOM 2022

Background: frequent pattern mining

- Traditional FPM workflow: gather-and-analyze



① Central algorithm & edge data

② Gather data

③ Central mining

Privacy in frequent pattern mining

- Access to raw data is restricted

> *Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject.*
>
> *—— EU General Data Protection Regulation (GDPR)*

- Privacy requirement: local differential privacy

Local differential privacy (LDP): *Any data publication mechanism M satisfies $\epsilon - LDP$ when for any raw local data d_i and d_j, and any possible output y,*

$$\Pr(M(d_i) = y) \leq e^{\epsilon} \Pr\left(M\left(d_j\right) = y\right)$$

FA schemes for privacy-preserving FPM

- RAPPOR[1]
  - Originally Designed for frequent item mining
  - Use bloom filter to encode the raw input
- SFP[2]
  - Designed for frequent sequence mining
  - Use count min sketches to encode raw input and sequence fragments
- TrieHH[3]
  - Designed for frequent string (essentially sequence) mining
  - Fail to satisfy LDP

[1] Erlingsson et al, RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response, CCS 2014

[2] Learning with Privacy at Scale, Apple Machine Learning Journal

[3] Zhu et al, Federated heavy hitters discovery with differential privacy, AISTATS 2020

FA schemes for privacy-preserving FPM

- Existing solutions cannot unify different FPM tasks
- Frequent sequence mining is naturally more difficult
  - SFP results in low performance
  - TrieHH can not satisfy local privacy requirement

	FPM scenario			Performance	
	Item	Itemset	Sequence	Data utility	Privacy
RAPPOR	✓	△		+++	✓
SFP			✓	+	✓
TrieHH			✓	++	△

Drawbacks of traditional solutions

- Why traditional FPM solutions results low data utility?
  - Significant noise is added on uploads to satisfy LDP
  - The uploads are indirect transformations of original data

- RAPPOR uploads a bloom filter with 128 bits
  - To satisfy $\epsilon = 2$, each bit should be flipped with probability 49.6%

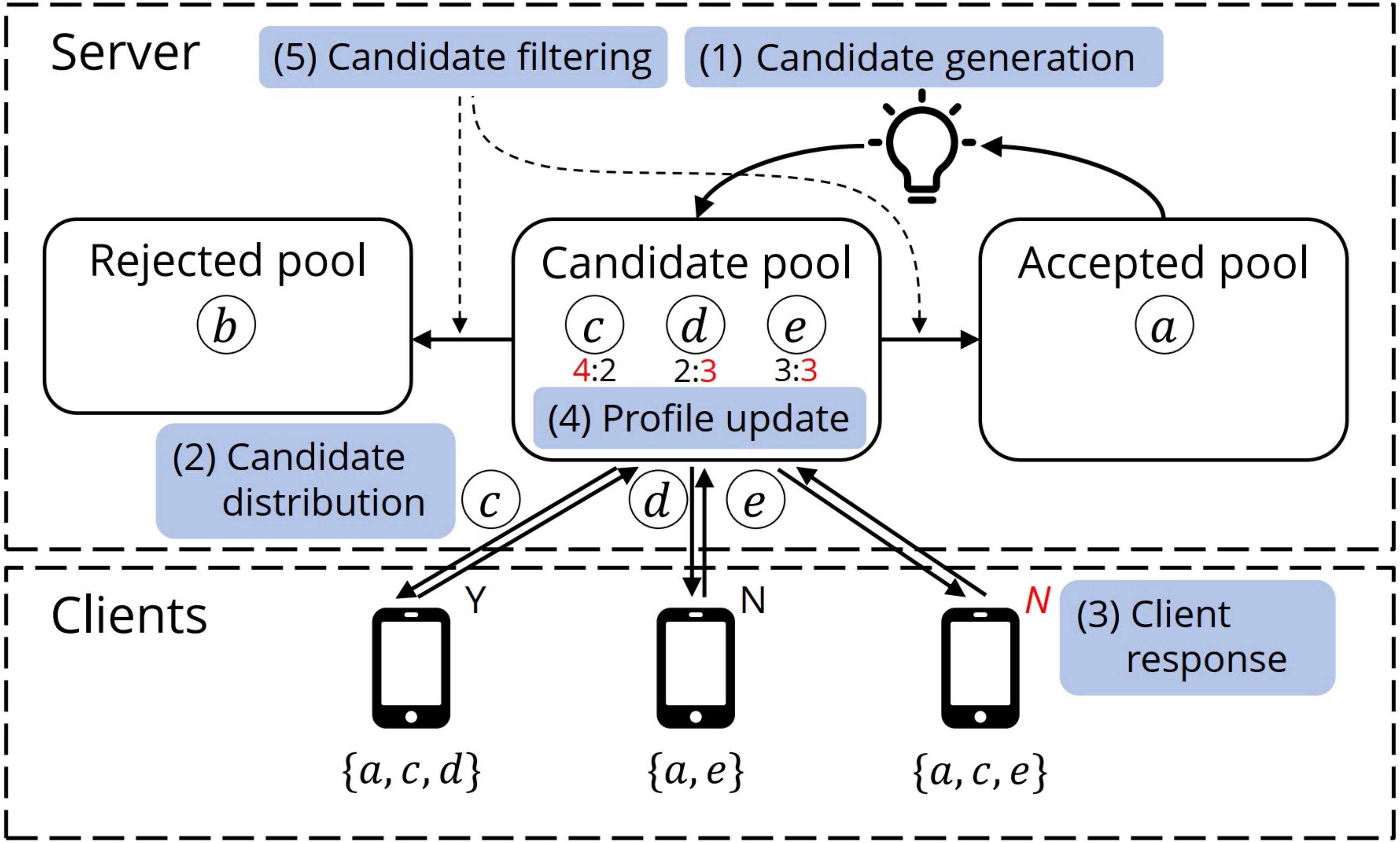
- The server is hard to recover raw data from the uploads
  - The uploads are bloom filters or count mean sketches

Our objectives

- Unify multiple FPM problems
- Satisfy strong privacy criteria (LDP)
- Achieve high data utility in all scenarios

	FPM scenario			Performance	
	Item	Itemset	Sequence	Data utility	Privacy
RAPPOR	✓	△		+++	✓
SFP			✓	+	✓
TrieHH			✓	++	△
Ours	✓	✓	✓	+++	✓

System design of FedFPM



Candidate generation

- The only module needs to be modified in different FPM tasks
- Generate new candidates based on those proven to be frequent

- Frequent item mining: enumerate all items in the first round
  - Candidate generation happens only in the first round
  - Other choices: increase granularity of items

Candidate generation

- Complex scenarios: Apriori property
  - The sub-structure of a frequent structure must be also frequent
  - Use smaller itemsets/sequences to generate longer ones

Frequent itemset mining

Accepted candidates

$$\{a, b, c\}, \{a, c, d\}$$

New candidate

$$\{a, b, c, d\}$$

Frequent sequence mining

Accepted candidates

$$(a \rightarrow c \rightarrow d) + (c \rightarrow d \rightarrow b)$$

New candidate

$$(a \rightarrow c \rightarrow d \rightarrow b)$$

- Significantly decrease the number of candidates to be verified

Steps 2~4

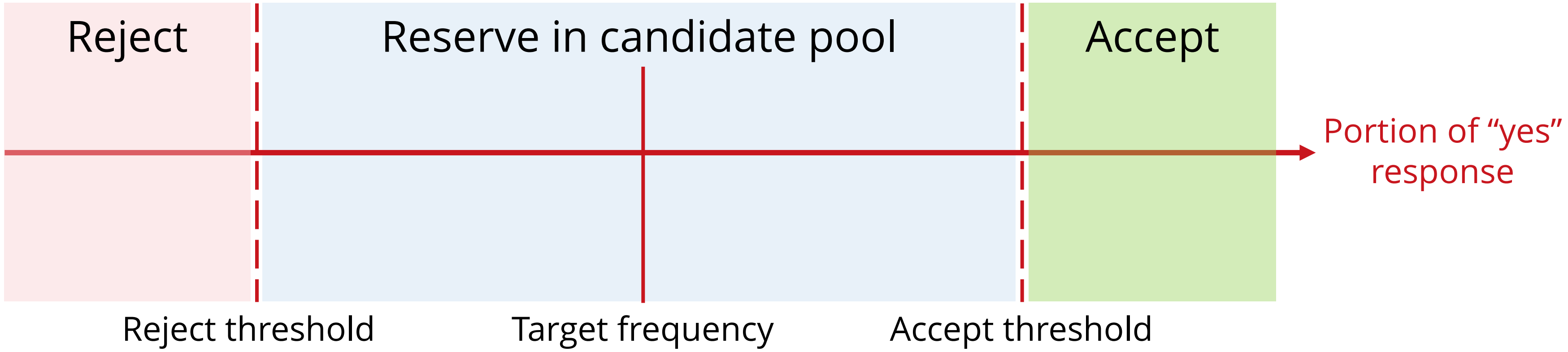
- **Candidate distribution:** randomly select one candidate to each participating clients

- **Client response:** The client checks whether the candidate presents in local data, gives a binary response, and flip it with possibility
  - The possibility is $1/(1+e)^{\epsilon}$ to satisfy LDP

- **Candidate profile update:** the server memorizes the numbers of yes/no responses for each client

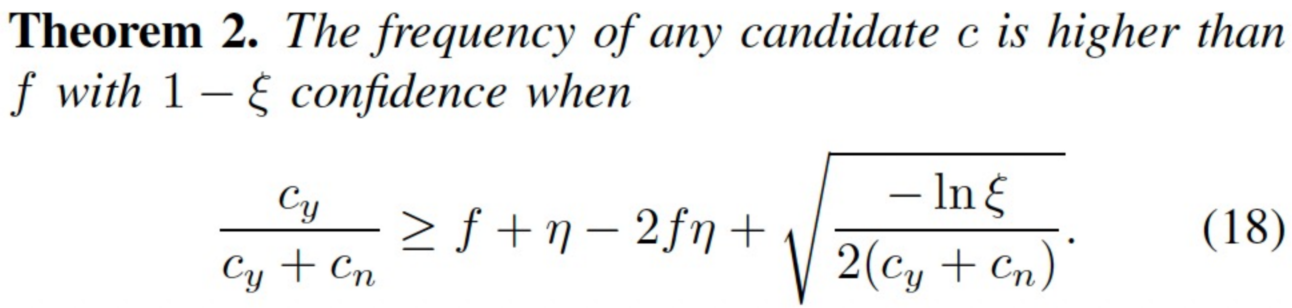
Candidate filtering

- Objectives: accept candidate or reject candidate
  - Whether a candidate is a frequent pattern with sufficient confidence?
  - Whether a candidate is not a frequent pattern with sufficient confidence?
- Two-sided filtering
  - Reject/accept a candidate with sufficient confidence
  - Otherwise, reserve it in candidate pool



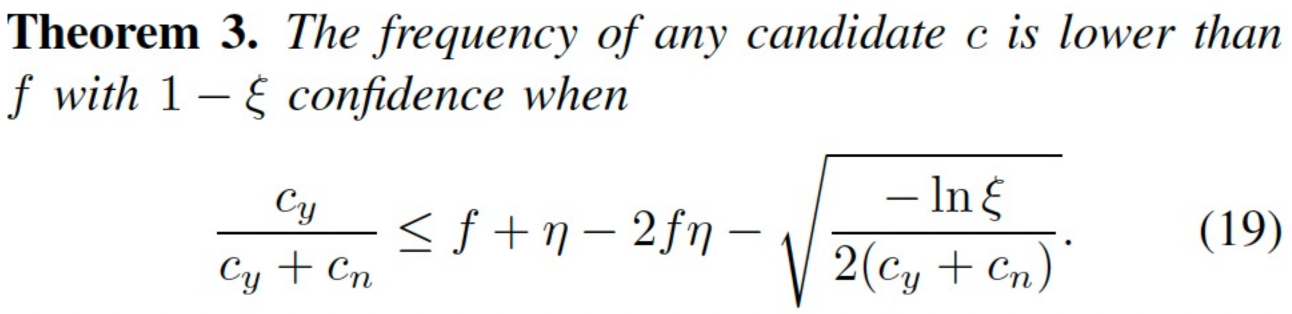
Candidate filtering

- Accept/reject thresholds are derived by the Hoeffding's inequality



Theorem 2. *The frequency of any candidate c is higher than f with $1 - \xi$ confidence when*

$$\frac{c_y}{c_y + c_n} \geq f + \eta - 2f\eta + \sqrt{\frac{-\ln \xi}{2(c_y + c_n)}}. \qquad (18)$$



Theorem 3. *The frequency of any candidate c is lower than f with $1 - \xi$ confidence when*

$$\frac{c_y}{c_y + c_n} \leq f + \eta - 2f\eta - \sqrt{\frac{-\ln \xi}{2(c_y + c_n)}}. \qquad (19)$$

Experiment settings

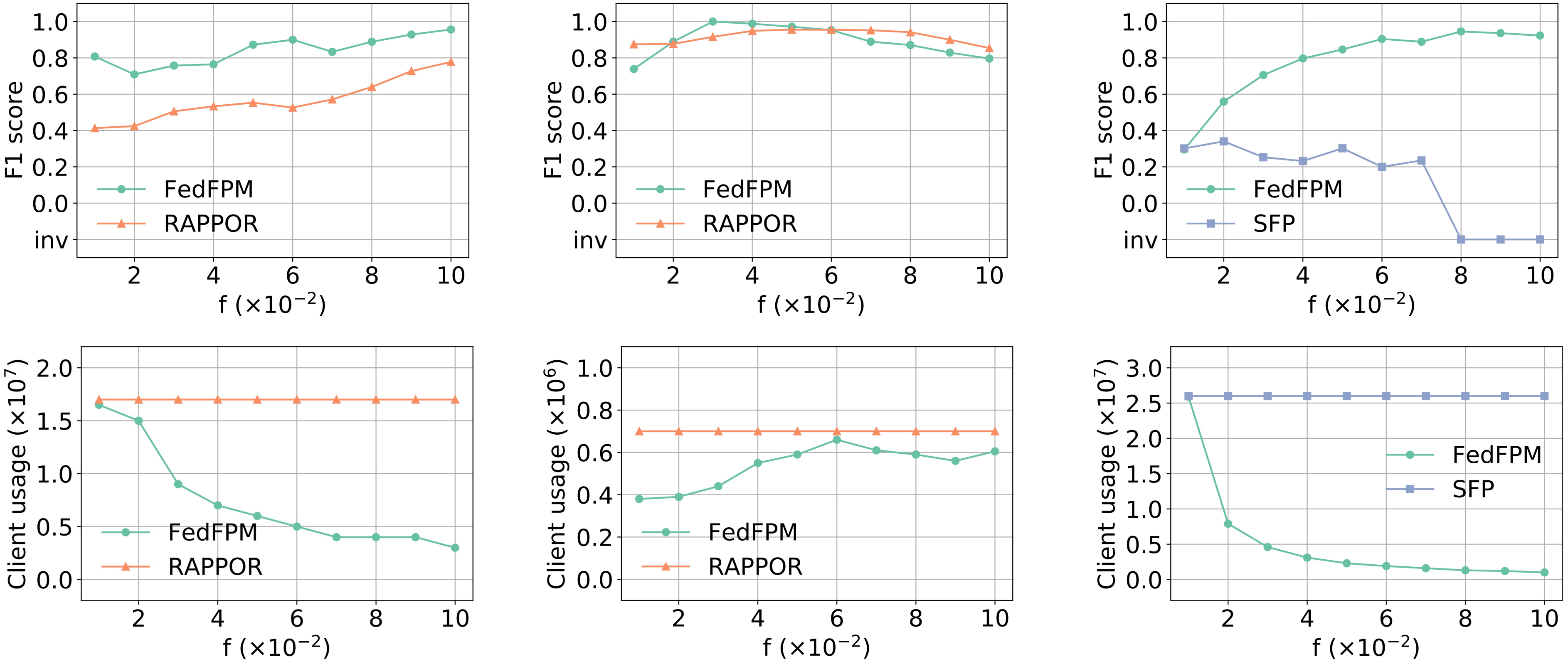
- Three datasets
  - *Kosarak* dataset for frequent item mining
  - *MovieLens* dataset for frequent itemset mining
  - *MSNBC* dataset for frequent sequence mining
- Two benchmarks
  - *RAPPOR* for frequent item/itemset mining (itemsets use one-hot encoding)
  - *SFP* for frequent sequence mining
- Metric: F1 score of FPM

Experiment settings

- Target FPM frequencies between 0.01 and 0.1
- LDP parameter $\epsilon = 2.0$
- Confidence of candidate filtering $\xi = 0.01$
- Each candidate receives as most $\kappa = 10^5$ responses

Evaluation results

- FedFPM gains higher F1 scores using less participating clients



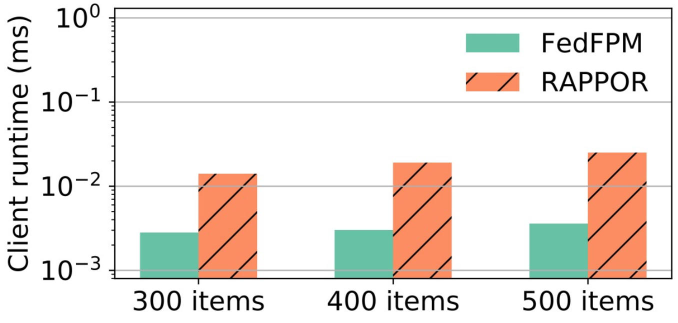
Frequent item mining

Frequent itemset mining

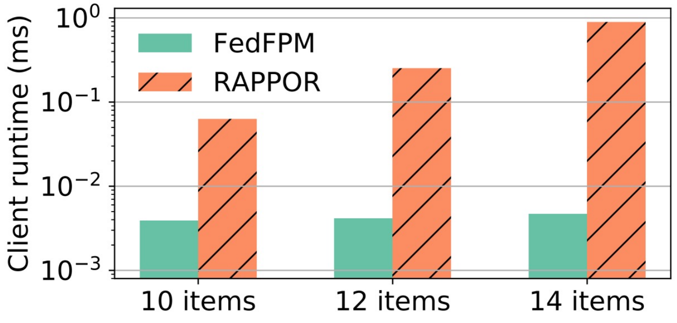
Frequent sequence mining

Evaluation results

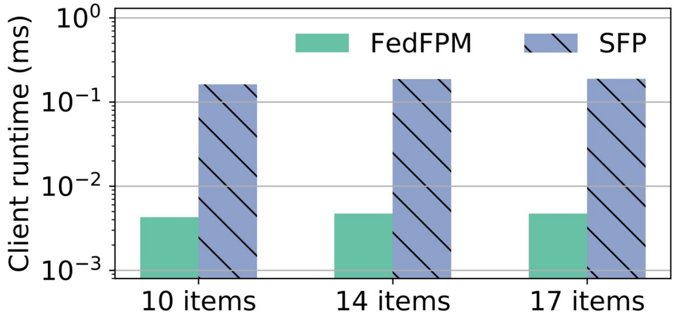
- FedFPM introduces much smaller client computation
  - Because it does not require clients to operate on hash tables
  - RAPPOR performs bad in frequent itemset mining



Frequent item mining



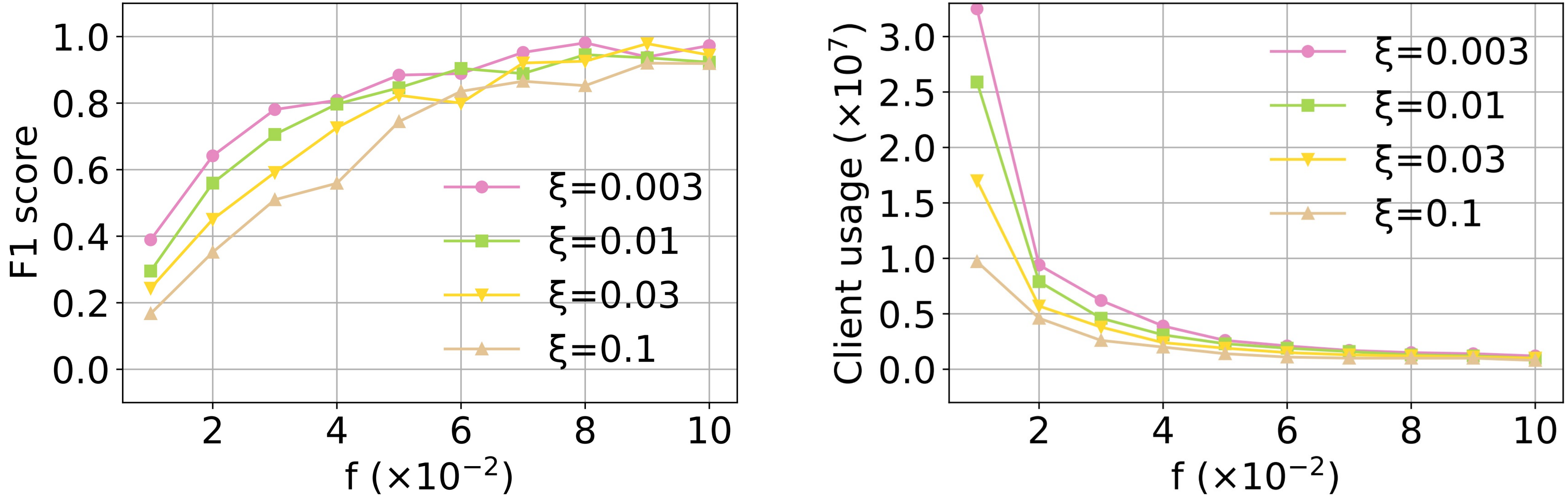
Frequent itemset mining



Frequent sequence mining

Evaluation results

- Trade-off: better data utility, but more participating clients
- Confidence of candidate filtering ξ acts as a handler



Frequent sequence mining

Summary

- Federated analytics-based frequent pattern mining scheme
  - Unify FPM tasks
  - Provide strong privacy preservation
  - Gain high data utility for complex patterns
- Key features
  - Apriori-based candidate generation
  - Two-sided candidate filtering
- Experiment results
  - Higher or similar F1 scores while using less clients
  - Less client computation time

Conclusions

- Introduction to Federated Analytics
  - Clarify its position in the research literature
  - Two triggers of federated analytics
    - Application demands
    - Technology Readiness
  - Federated video analytics: a first example on federated analytics

- Opportunities and challenges

- Federated analytics examples
  - Federated analytics for privacy-demanding systems
  - Federated analytics for enhancing privacy-preserving systems

Thank you!

Q&A

Email: dan.wang@polyu.edu.hk
si-ping.shi@connect.polyu.hk