Network Traffic Classification with Federated Learning

Presenter: Dan Wang

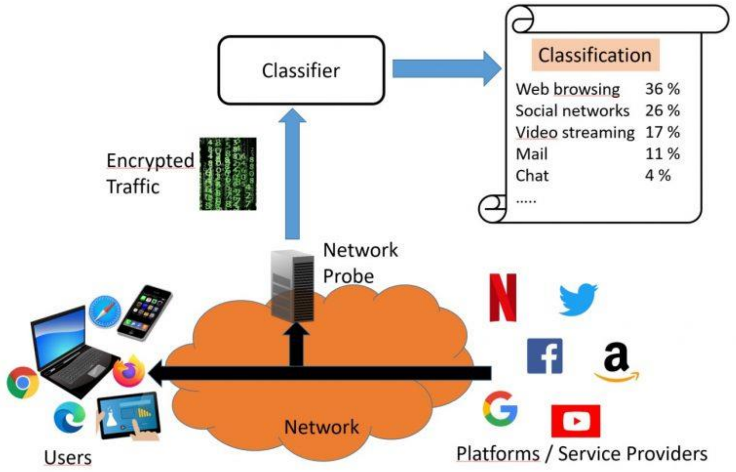
Department of Computing
The Hong Kong Polytechnic University

Outline

- Background
  - Network Traffic Classification
  - Federated Learning
- Federated Learning for Network Traffic Classification
  - A Federated Approach for Network Traffic Classification in Heterogeneous Environments
  - Robust Federated Learning for Network Traffic Classification with Noisy Labels
- Conclusion

Background

- Network Traffic Classification
  - Identifying the type or class of traffic flowing over a network
  - A foundation for many network security and network management applications
  - Applications: traffic engineering, network monitoring, Quality of Service



Background

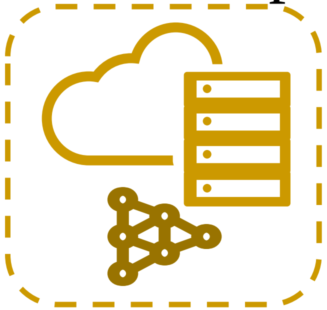
- Methods
  - Traditional network traffic classification methods
    - Port-based methods
    - Payload-based methods

Less effective for dynamic port numbers or encrypted data

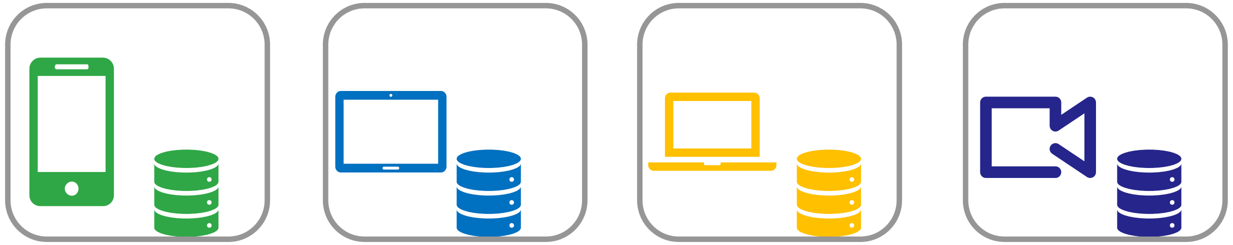
  - Deep learning
    - A large amount of labelled traffic data is required for learning
    - Privacy leakage risk of raw data in each client
      - i.e., traffic data related to the user behavior
    - Lack of scalability
      - Transferring all this data to a central server for processing can be inefficient and may not scale well

Background

- Federated learning
  - Introduced by Google in 2017
  - A promising learning paradigm proposed to protect user data privacy
  - Collaboratively learn a model while keeping all the data in local
    - Global model distribution

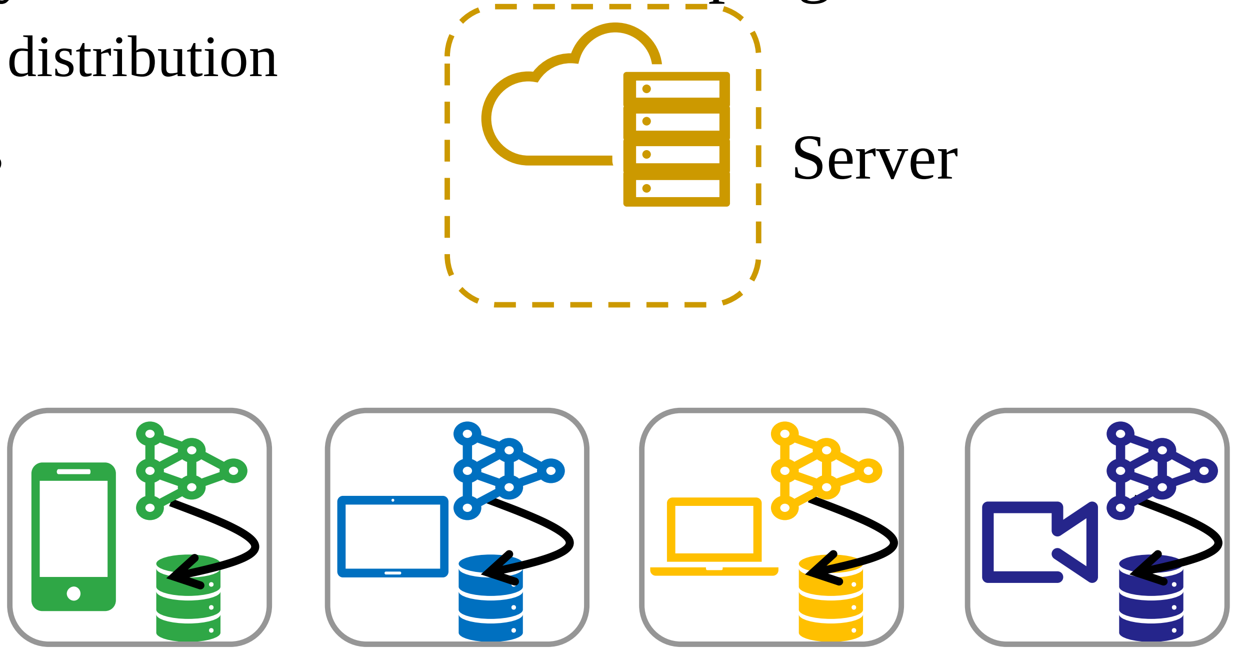


Server



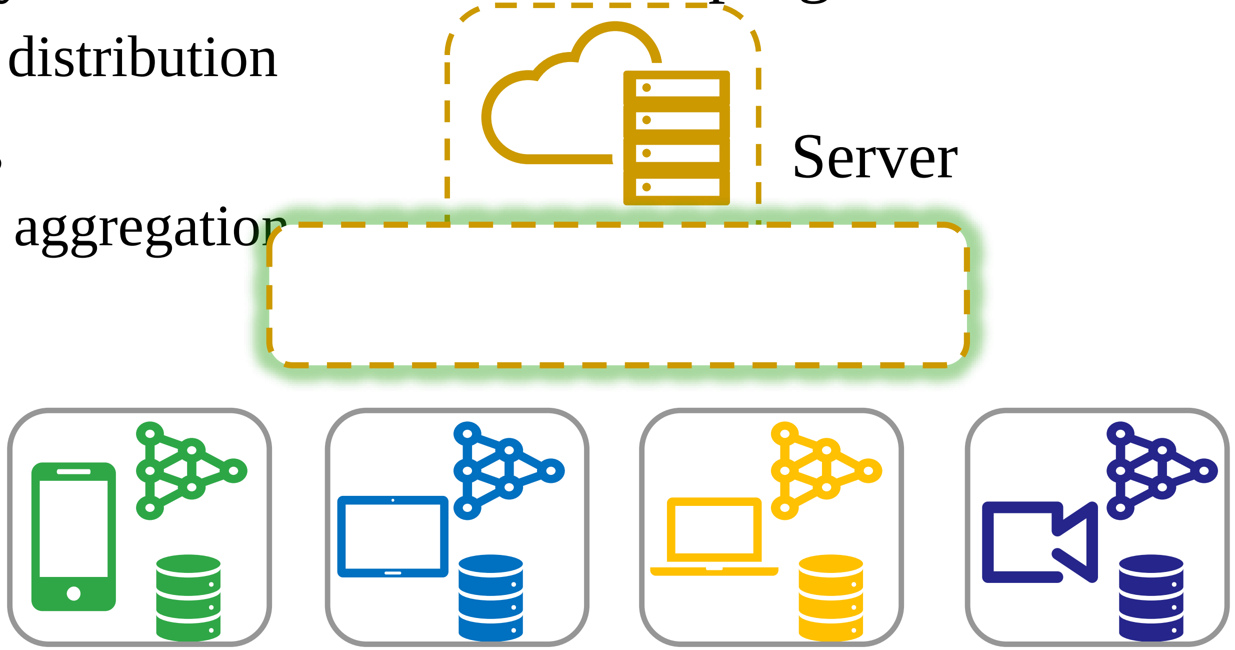
Background

- Federated learning
  - Introduced by Google in 2017
  - A promising learning paradigm proposed to protect user data privacy
  - Collaboratively learn a model while keeping all the data in local
    - Global model distribution
    - Local training



Background

- Federated learning
  - Introduced by Google in 2017
  - A promising learning paradigm proposed to protect user data privacy
  - Collaboratively learn a model while keeping all the data in local
    - Global model distribution
    - Local training
    - Local updates aggregation



Repeat these process until convergence

Challenges of Applying Federated Learning

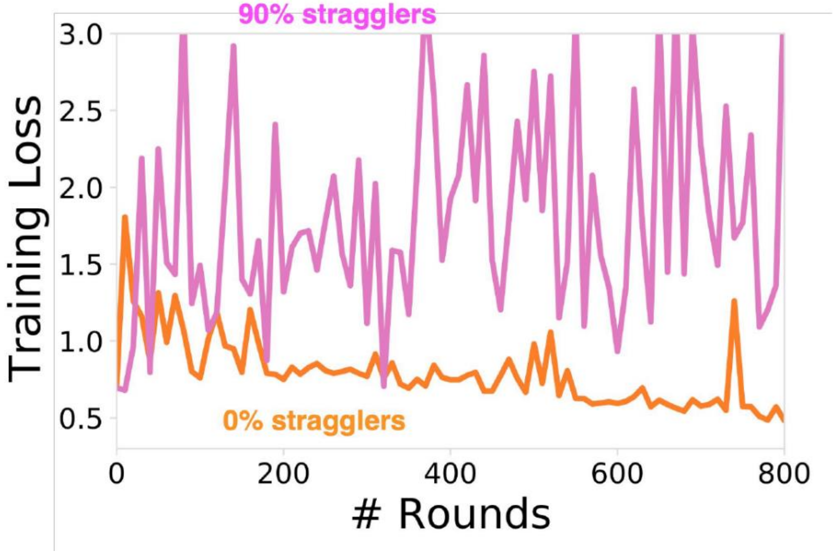
- Heterogeneity **First work**
  - Participating clients can have significant differences in terms of their *computational resources*, *network connectivity*, *availability* , and *the amount of the data* they have.
- Resilience to noisy data **Second work**
  - Data noise (i.e., noisy labels) occurred during learning
- Communication Overhead
  - Frequent communication between the central server and the client devices
- System Design and Management
  - Coordinating across many devices, handling device failures or dropouts

A Federated Approach for Network Traffic Classification in Heterogeneous Environments

Heterogeneous Environments

- 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]

Heterogeneous Environments

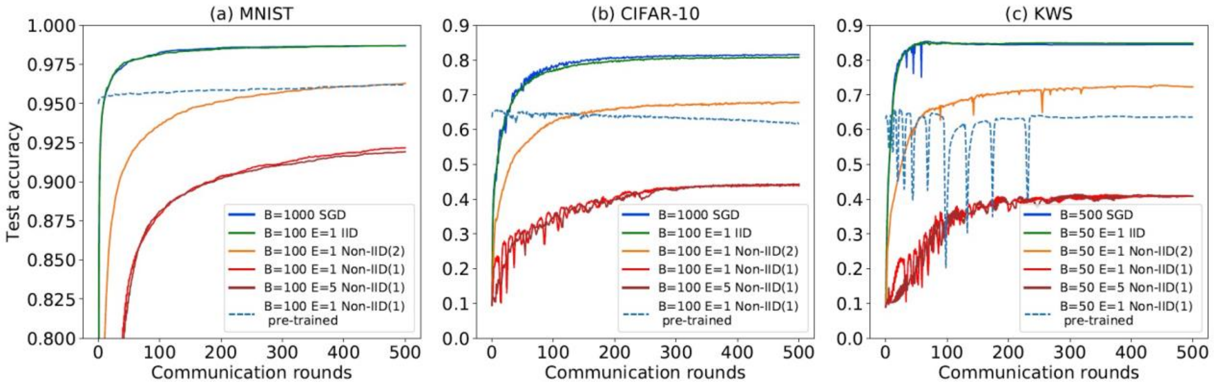
- Data Heterogeneity
  - Class distribution of the client data is skewed

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

TABLE I
CLASSIFICATION ACCURACY WITH FL IN HETEROGENEOUS ENVIRONMENTS

IID Environment	Non-IID Environment	
	Low Heterogeneity	High Heterogeneity
95.5%	88.0%	83.7%

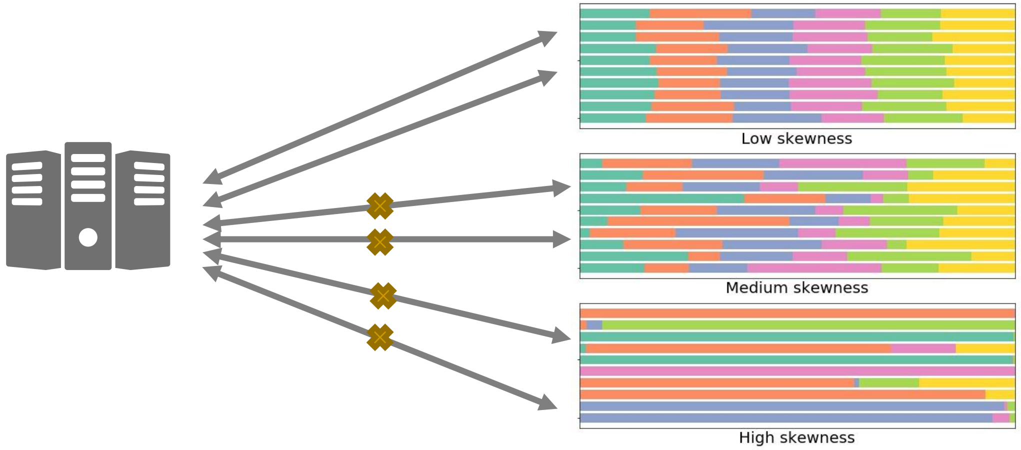
SGD: centralized training
IID: independent and identical distributed
Non-IID (1): each client owns data of 1 class
Non-IID (2): each client owns data of 2 classes



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

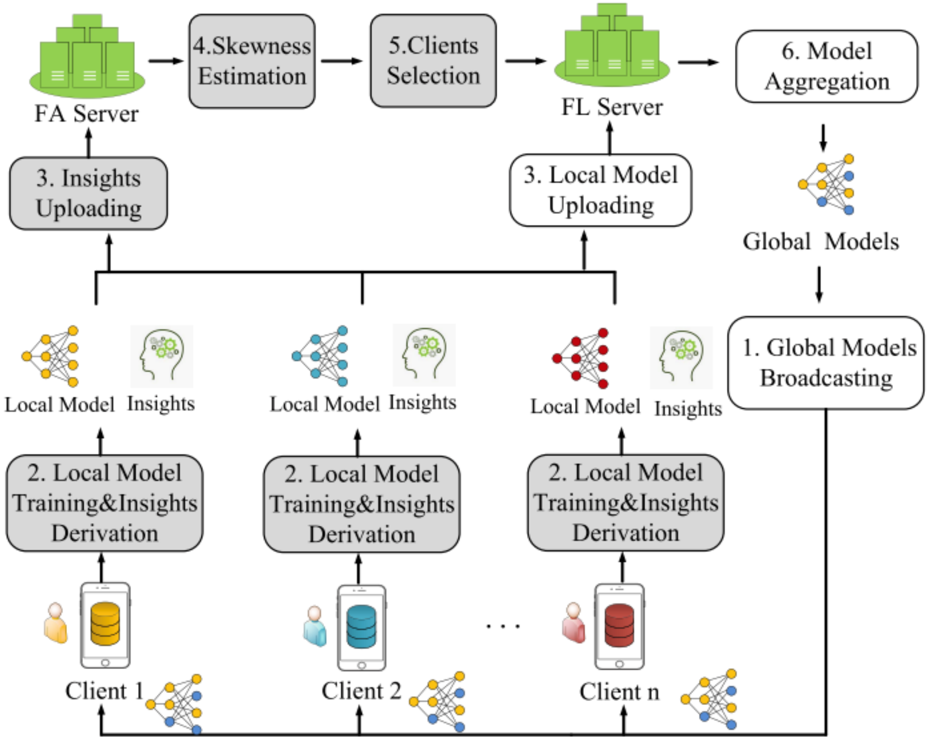
FEAT: A Federated Approach for Network Traffic Classification in Heterogeneous Environments

- Motivation
  - Clients with different skewness are *not equally beneficial* to federated learning
    - Skewness: the *severity* of data heterogeneity
- Idea: heterogeneity-aware client selection
  - Measure the skewness of the clients
  - Select clients with low skewness



Heterogeneity-aware Client Selection

- Three steps
  - Insight generation
    - Clients generate insights about the skewness of its local data
  - Skewness generation
    - Server aggregates the insights and infer about client skewness
  - Client selection
    - Server selects the participating clients based on the skewness estimation



Heterogeneity-aware Client Selection

- Challenges

Step 1: Insight generation	Step 2: Skewness estimation	Step 3: Client selection
• The insight should be informative about the client skewness • The insight should be indirect to protect raw data privacy	• It should derive useful knowledge from the indirect insights • The procedure should be mathematically sound	• The selection should be robust to the system uncertainty • The selection should satisfy requirements of the host tasks

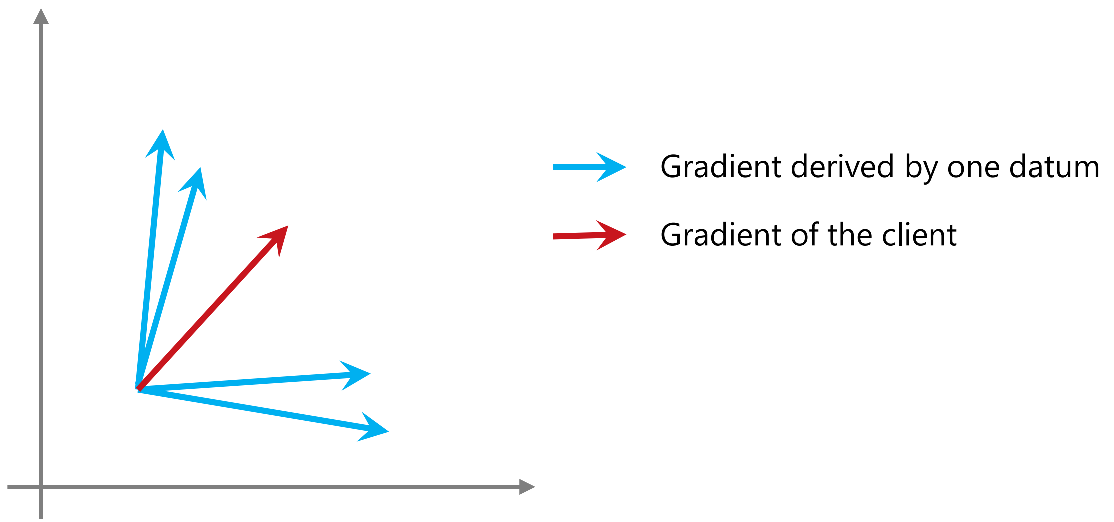
Step 1: Insight Generation

- 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-bandit dueling problem
  - Dueling bandit design
    - Participating client "duel" with each other using their rewards
    - Train the bandit using the dueling results

$R_i = -2$ | $R_j = -3$ | $R_k = -10$

$win = 2, lose = 0$ | $win = 1, lose = 1$ | $win = 0, lose = 2$

Step 3: Client Selection

- Client selection is formulated as a multi-bandit dueling problem
  - Thompson Sampling based Clients Selection
    - 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
  - λ is designed for the tradeoff between selecting the clients with low skewness and providing the training model with more raw traffic data samples.

Theoretical Analysis

- Convergence Analysis
  - The distance of the loss value between the learned model and the optimal model is bounded

Theorem 2. *Let* $\kappa = (L/\mu), \rho = max\{8\kappa, e\}$ *and the learning rate* $\eta_t = (2/\mu(\rho + t))$. e *is the local update epoch.* r_k *is the weight of client* k. *Then, FEAT satisfies*

$$E[F(\bar{w}_t)] - F^* \leq \frac{\kappa}{\rho + t}\left(\frac{2(P+Q)}{\mu} + \frac{\mu(\rho+1)}{2} E\|w_1 - w^*\|^2\right)$$

where

$$P = \Sigma_{k=1}^{N} r_k^2 \sigma_k^2 + 6L\Gamma + 8(e-1)^2 G^2, Q = \frac{4}{d} e^2 G^2$$

$$F = \sum_{k=1}^{N} r_k F_k, \Gamma = F^* - \Sigma_{k=1}^{N} r_k F_k^*.$$

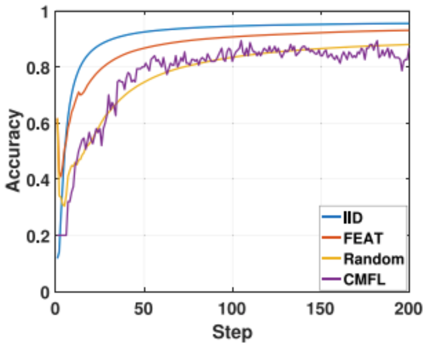
Here, F^* *and* F_k^* *denote the minimal value of* F *and* F_k*, respectively.*

Evaluation

- Setup
  - Dataset
    - QUIC: contains traffic data from five Google Services
    - ISCX: contains traffic data from 31 applications
  - Heterogeneous Environment Setting
    - Low heterogeneity: Dirichlet distribution with α uniformly sampled from [0, 0.2] and [0.2, 3]
    - High heterogeneity: Dirichlet distribution with α uniformly sampled from [0, 0.1] and [0.1, 5]
  - Benchmarks
    - IID: upper bound baseline
    - Random: random clients selection
    - CMFL: client selection method that is based on sign counts
    - WCL: select the clients based on their loss values

Evaluation

- Results
  - FEAT can improve the traffic classification accuracy to 68.6% in the environment with high heterogeneity compared to benchmarks
  - FEAT can speed up the convergence by 2.6× and 1.9× compared to benchmarks



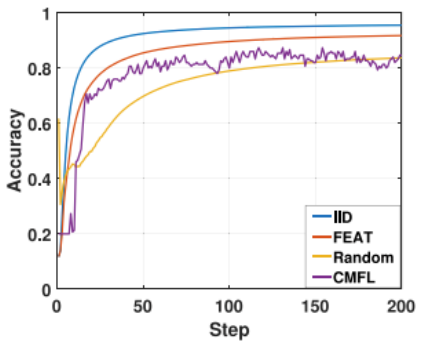


Fig. 3. Accuracy under environments with different heterogeneities on QUIC. (a) Low heterogeneity. (b) High heterogeneity.

TABLE II
COMPARISON OF ACCURACY UNDER DIFFERENT METHODS ON QUIC

Heterogeneity	Methods	Accuracy (%)	Improvement (%)
Low	IID	95.5	100.0
	Random	88.0	0.0
	CMFL	86.7	-17.3
	FEAT	93.0	66.7
High	IID	95.5	100.0
	Random	83.7	0.0
	CMFL	84.7	9.1
	FEAT	91.8	68.6

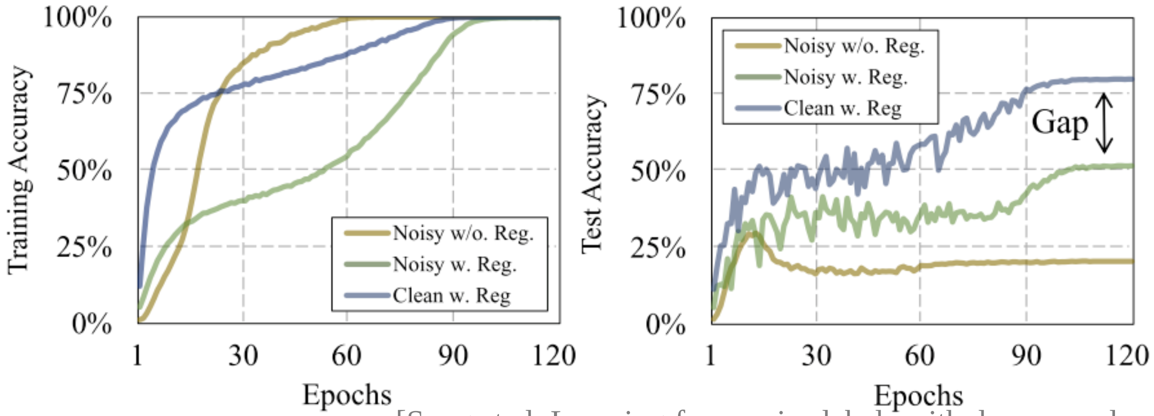
TABLE III
COMMUNICATION ROUNDS NEEDED TO REACH 80% OF TARGET ACCURACY UNDER DIFFERENT METHODS

Heterogeneity	Methods	Rounds to 80%	Speedup
Low	IID	12	1.8x
	Random	56	-2.6x
	CMFL	41	-1.9x
	FEAT	22	1.0x
High	IID	12	1.9x
	Random	79	-3.4x
	CMFL	40	-1.7x
	FEAT	23	1.0x

Robust Federated Learning for Network Traffic Classification with Noisy Labels

Noisy Labels in Network Traffic Classification

- Sources of Noisy Labels
  - Non-expert labeling
  - The existence of background unknown traffic flow during collection
    - i.e., the traffic of a new application
- Impact of Noisy Labels
  - Severely degrading the performance of learned model



[Song et al. Learning from noisy labels with deep neural networks: A survey. IEEE TNNLS 2022.]
[Anderson et al. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. ACM SIGKDD 2017]

Noisy Labels in Network Traffic Classification

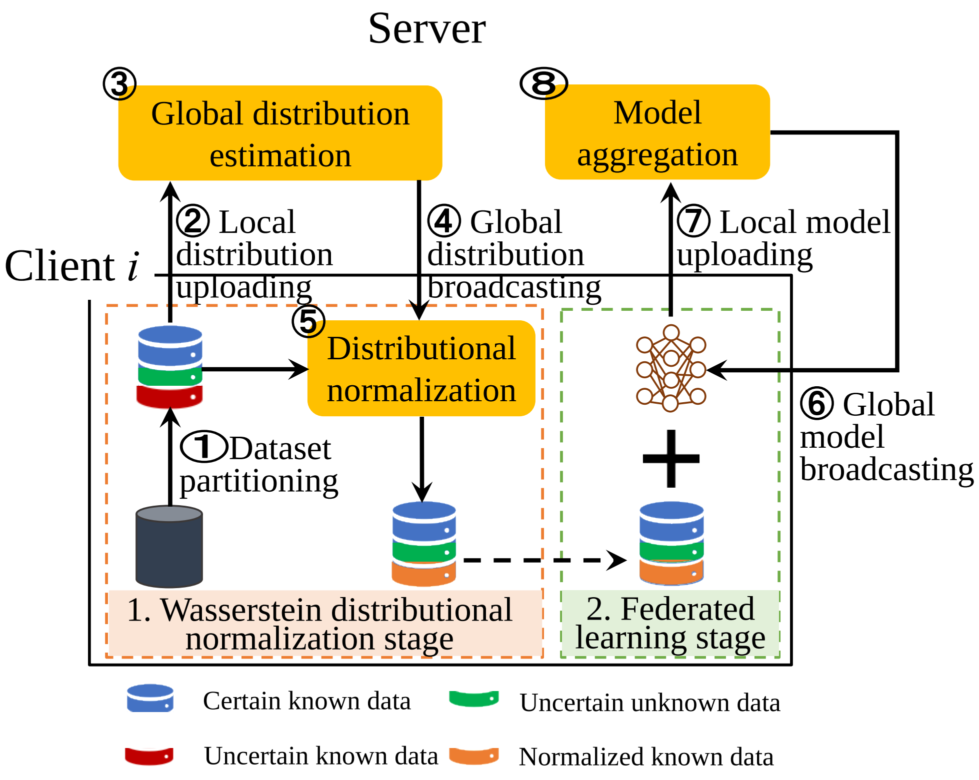
- Existing Noise Elimination Methods
  - Noise is detected and removed from the training process
  - Simple to apply and perform well for data centres (i.e., Internet Service Providers (ISPs)) with a large amount of traffic data
- Limitations
  - May lead to poor performance of the learned network traffic classifier for mobile devices which generate a relatively small amount of traffic data
  - Privacy leakage risk
    - All the local traffic data is required to be collected to a central server for noise detection.

Distributionally Robust Federated Learning for Network Traffic Classification with Noisy Labels

- Motivation
  - The data feature of the noisy labelled traffic data is clean
  - The underlying true distribution of the noisy labeled data is statistically close to the clean traffic data
- Idea: Wasserstein Distributionally Normalization
  - Transform noisy labeled data to be close to the clean traffic data
  - Jointly take the transformed noisy traffic data and the clean traffic data into training

Wasserstein Distributionally Normalization

- Three steps
  - Local dataset partitioning
    - Partition the local traffic data in each client into certain clean data and uncertain noisy data
  - Global clean data distribution estimation
    - Estimate the global clean data distribution based on the uploaded local data distribution
  - Distributional normalization
    - Normalize the uncertain noisy data to be close to the clean data distribution



Step 1: Local dataset partitioning

- Small-loss criteria
  - The loss value of a noisy labeled data sample is larger than a clean data sample
    - Smaller the loss value of a data sample, the higher the probability of being clean
  - Let ζ be the loss threshold, and $\mathcal{D}^c$ be the certain clean data set

$$\mathcal{D}^c = \{(x, y) | \ell(x, \theta, y) \leq \zeta; (x, y) \in \mathcal{D}\},$$

Step 2: Global Clean Data Distribution Estimation

- Federated distribution estimation
  - The local clean traffic data is located in each client and can not be sent to the server due to privacy concerns
  - Each client estimates the local distribution b_i of the certain traffic dataset $\mathcal{D}_i^c$ and sends it to the server
  - The server constructs the virtual observations according to the local distributions and then estimates the global distribution $g(e)$

$$g(\boldsymbol{e}) = \sum_{i=1}^{N} w_i \psi_i(\boldsymbol{e}_i),$$

Gaussian kernel

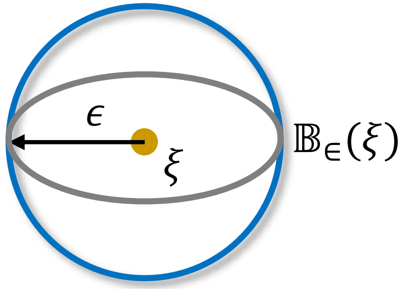
  - We leverage the Markov Chain Monte Carlo with a delayed rejection to solve the problem

Step 3: Distributional Normalization

- Wasserstein certified robust region construction
  - A ball of radius ϵ around the certain clean traffic data distribution ξ

Definition. (Wasserstein certified robust region) Let $\mathcal{P}_2$ be the distribution space. We define the certified robust region $\mathbb{B}_{\xi}(\epsilon)$ in this space as follows:

$$\mathbb{B}_{\xi}(\epsilon) = \{\varsigma \in \mathcal{P}_2 : W_2(\varsigma, \xi) \leq \epsilon\}$$



  - Each probability distribution in the certified robust region is statistically close to the probability distribution of the certain clean traffic data set

Step 3: Distributional Normalization

- Distributional normalization function specification
  - The normalization function $\mathcal{F}$ should ensure the normalized probability distribution $\hat{\omega} = \mathcal{F}(\omega)$ is lying in the certified robust region $\mathbb{B}_{\xi}(\epsilon)$

$$\sup_{\mathcal{F}(\omega)} \mathcal{W}_2(\mathcal{F}(\omega), \xi) \leq \epsilon.$$

Steepest decent direction to maximize the distance

  - $\mathcal{F}$ is defined as the *gradient flow* in Wasserstein-2 space

Definition. (Wasserstein normalization function) Let $\mathcal{F}$ be the distributional normalization function which transforms probability distribution ω to ω_t, and $\mathcal{F}_{\mathrm{t}}(\omega) = \omega_t$. We define $\mathcal{F}$ as a gradient flow in the Wasserstein-2 space and ω_t satisfies the following continuity equation :

$$\frac{\partial \omega_t}{\partial_t} = \nabla \cdot (\omega_t v_t)$$

where $d\omega_t = p_t d\mathcal{N}_\xi, d\mathcal{N}_\xi = dq_t dx, v_t = \nabla \log q_t$. Here, p_t and q_t are probability density functions, and $\mathcal{N}_\xi$ is a Gaussian distribution with mean $\mathbf{m}_\xi$ and covariance Σ_ξ.

Step 3: Distributional Normalization

- Distributional normalization function specification
  - The gradient flow in the Wasserstein-2 space is also the Fokker-Planck equation

$$\frac{\partial \rho(t,x)}{\partial t} = \nabla \cdot (\rho(t,x)\nabla V(x)), \quad \rho(0,x) = \rho_0(x).$$

  - Obtaining the normalized data distribution by solving the following stochastic differential equation (SDE)

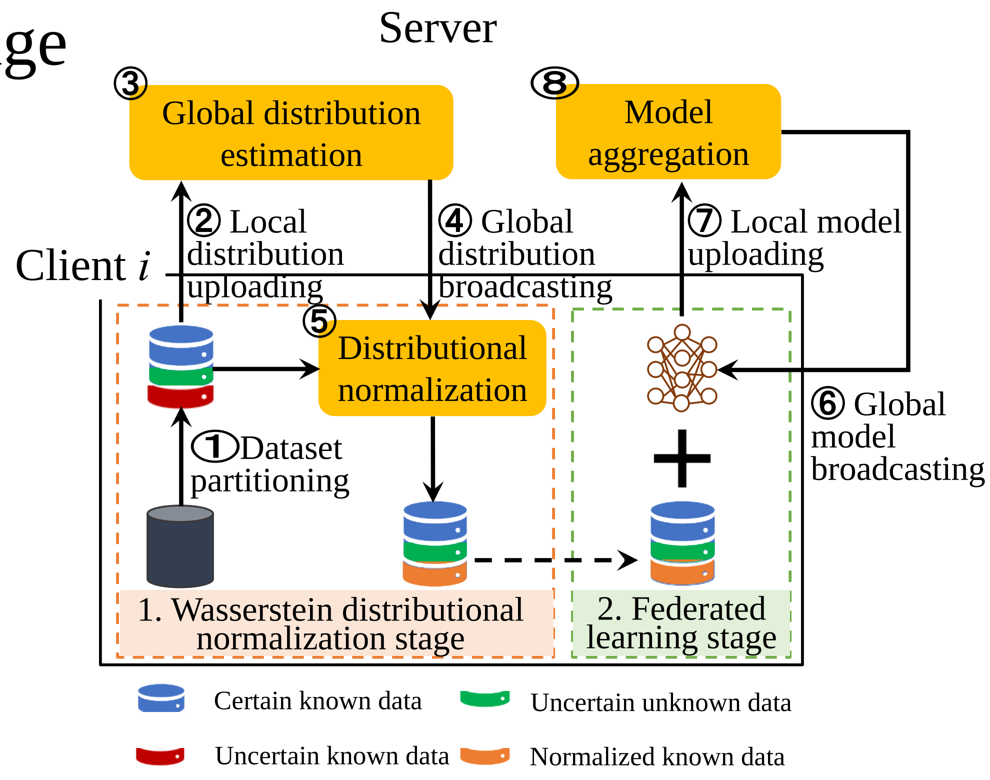
$$dX_t = -\nabla\phi\left(X_t; \mathbf{m}_\xi\right) dt + \sqrt{2\tau^{-1}\Sigma_\xi}\, d\mathbf{W}_t, \quad X_0 \sim \rho_0,$$

    - Euler-Maruyama scheme can be used to simulate the stochastic process X_t

$$X_{t+1} = X_t - \nabla\phi\left(X_t; \mathbf{m}_\xi\right) \Delta_t + \sqrt{2\tau^{-1}\Delta_t\Sigma_\xi}\, Z,$$

Robust Federated Network Traffic Classifier Learning Algorithm

- RFNTC algorithm: two-stage learning
  - Wasserstein distributional normalization stage
  - Federated learning stage



Theoretical Analysis

- Concentration Analysis
  - The noisy labeled uncertain traffic data is proved to be normalized to the certified robust region

Lemma 1. *Let Assumption 1 holds, and π is the Lipschitz constant of softmax function s. There exists a constant σ satisfy the following probability inequality:*

$$\mathcal{F}_T(\omega)\big(\{z: |s(X_T(z)) - \mathbb{E}_\xi[s]| \geq \sigma\}\big) \leq 6e^{-\frac{2C^{\frac{3}{2}}}{\sqrt{K_2}}},$$

where ω and ξ denote the uncertain and certain probability distributions, respectively, and $C = \frac{\sigma}{\pi}$.

Theoretical Analysis

- Robustness Analysis
  - The distance of the loss value between the learned model and the optimal model is bounded

Theorem 1. *Let Assumptions 1 to 5 hold and E is the number of local iterations. Let $\kappa = \frac{L}{\mu}, \gamma = max\{8\kappa, E\}$ and $\Delta_0 = \mathbb{E}\|\theta_0 - \theta^*\|^2$. We have*

$$\mathbb{E}[\ell(\theta_K)] - \ell^* \leq \frac{\kappa}{\gamma + K - 1}\left(\frac{2B}{\mu} + 4L\Delta_0\right),$$

where

$$B = \sum_{i=1}^{N} \frac{\sigma_i^2}{N^2} + 6L\Gamma + 8(E-1)^2 G^2$$

Evaluation

- Setup
  - Dataset
    - ISCXVPN2016: There are 17 applications belonging to 7 application categories in this dataset, and we pre-process the PCAP format traffic data with CICFlowMeter tool.
  - Traffic Classification Model
    - A CNN-based network traffic classifier
  - Benchmarks
    - FedAvg (AVG): baseline
    - ROLC-NC-D: a centralized robust traffic classification method
    - ROLC: a federated version of ROLC-NC-D

Evaluation

- Results
  - The proposed RFNTC algorithm can improve the accuracy of the learned model for up to 1.05 times compared to benchmarks

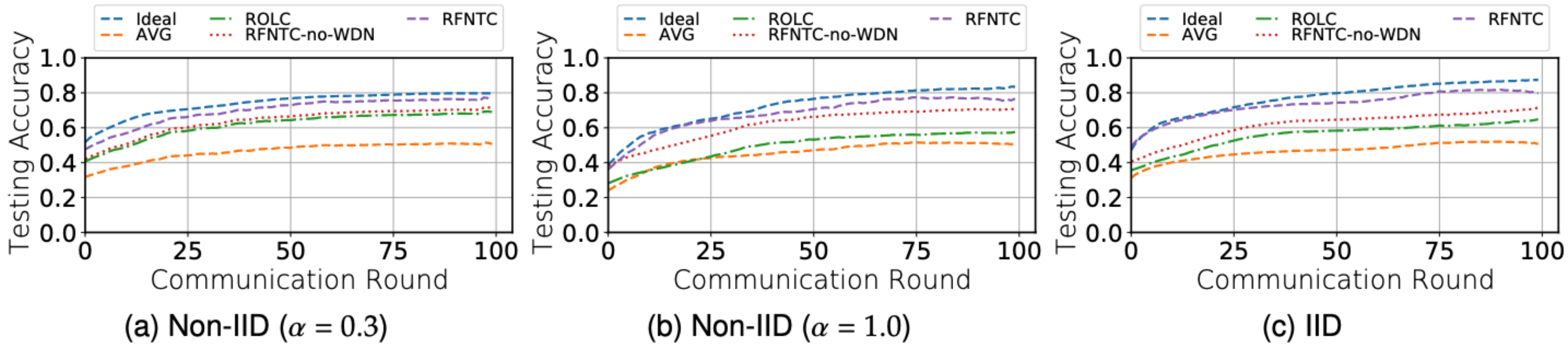


Fig. 2: The accuracy of the learned network traffic classifier with different training methods.

Evaluation

- Results
  - The proposed RFNTC algorithm improves the accuracy of the learned classifier by 0.5 times even when a large noisy clients ratio occurs (i.e., the fraction of noisy clients is 0.5),

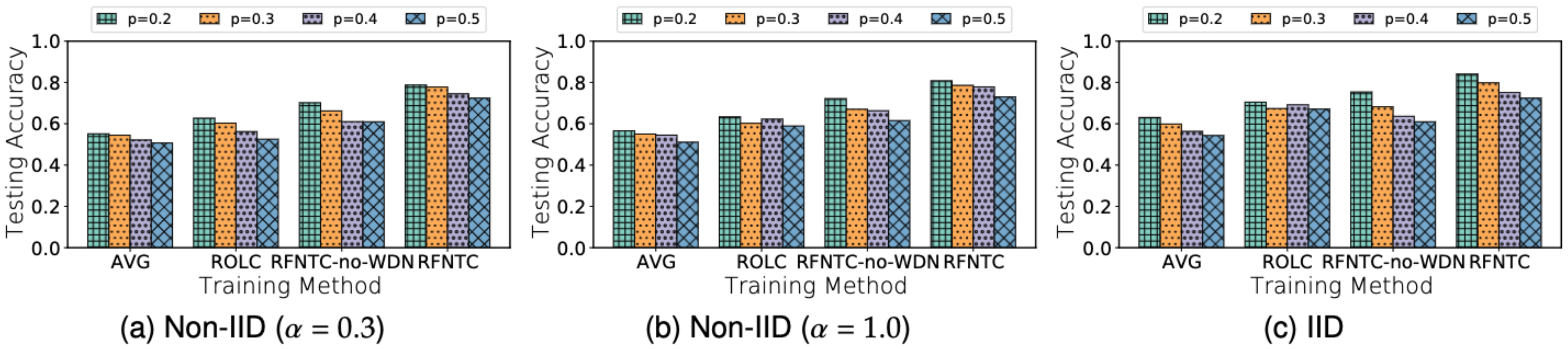


Fig. 3: Top-1 accuracy of different training methods with various noisy client ratios (from 0.2 to 0.5).

Conclusion

- Federated learning is a promising paradigm for traffic classification
- FEAT: network traffic classification in heterogeneous environment
  - Theoretically guaranteed Skewness estimation: Hoeffding's Inequality
  - Robust client selection: dueling bandit and quality & quantity parameter
- RFNTC: network traffic classification with noisy labels
  - Privacy-preserving global distribution estimation: federated analytics
  - Theoretically guaranteed distribution normalization: Wasserstein distributional normalization
- Extensive evaluation results present the superior performance of the proposed methods

Thank you!

Q&A

Email: dan.wang@polyu.edu.hk