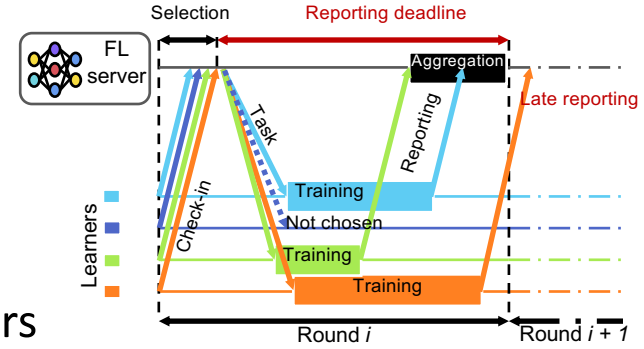
Practical and Efficient Federated Learning

Ahmed M. A. Sayed
Associate Professor @ School of EECS
ahmed.sayed@qmul.ac.uk
https://eecs.qmul.ac.uk/~ahmed

RECAP - Federated Learning Life-cycle



- FL server coordinates the learning stages

1. Online Learners **check-in** with the server
2. Participants **Selection** Stage:
   - The server selects a pre-set target number of learners
3. Participants **Local Training** Stage:
   - The selected participants perform the training task on the local dataset
4. Participants **Upload Updates** Stage:
   - The participants completing the task within deadline upload their updates.
5. Server **Aggregation** Stage:
   - The server aggregates the updates to produce the new global model

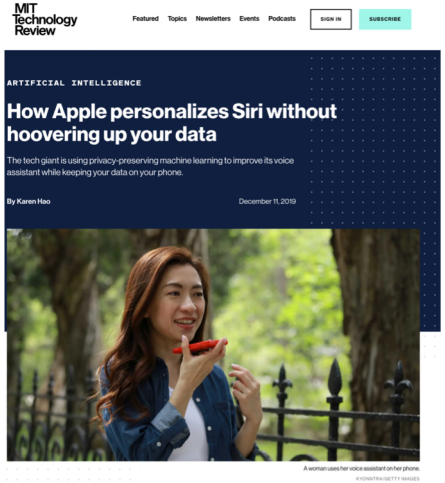
Why Federated Learning Evolved

In Many AI/ML-based applications

- User Data is **Distributed & Siloed**!
- User Data is **private & not shared!**

- Internet of Things (IoT)
- Healthcare
- Finance
- Industry
- Smart-city/grid
- Telecommunications
- Self-driving vehicles
- ….

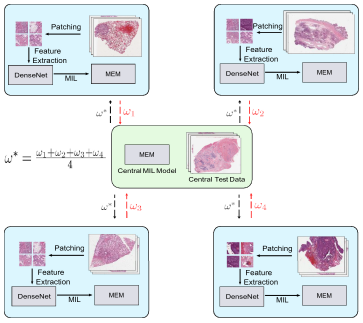
Apple: Voice recognition



Gboard next-word prediction

Using FL, better next- word prediction accuracy: +24%

A. Hard, et al. Federated Learning for Mobile Keyboard Prediction. arXiv:1811.03604



Medical Imaging

Adnan, M., et al. Federated learning and differential privacy for medical image analysis. Sci Rep 12, 1953 (2022). Nature 2022.

Major Challenges in FL & Outline

- **Data / Convergence** / **Systems / Incentive** Aspects of FL:
  - **Part 1 focused on this,** and mitigation methods have been discussed

1. **Resource** Heterogeneity :
   - Clients' devices configuration are *heterogeneous (*compute, network, etc).
     - Some devices become stragglers or become unavailable to contribute
2. **Energy** Heterogeneity
   - FL performance relies on learners submitting updates.
     - Low-energy devices fail to contribute, giving an edge to powerful ones to dominate the model.
3. **Behaviour** heterogeneity:
   - Some clients are either *malicious* or *malfunctioning*, impacting global model

REFL: Resource-Efficient Federated Learning

ACM EuroSys 2023

https:// dl.acm.org/doi/10.1145/3552326.3567485

CODE: https://github.com/ahmedcs/REFL

Status Quo

- Many FL systems aim to improve the quality of FL trained models.
- Goal -> Reduce Time to Accuracy: Reduce Time ↓ + Improve Quality ↑

FedProx (MLSys 2020), Yogi (ICLR 2021)

- Improve the statistical efficiency of the learning process

Oort (USENIX OSDI 2021)

- Bias the client selection to reduce the training time by exploiting the fast learners.

SAFA (IEEE ToC 2021)

- Invokes all learners for training and allows semi-synchronous model updates to boost the statistical efficiency

FLEET (ACM Middleware 2021)

- Boost statistical efficiency via asynchronous updates with damping and boosting rules.

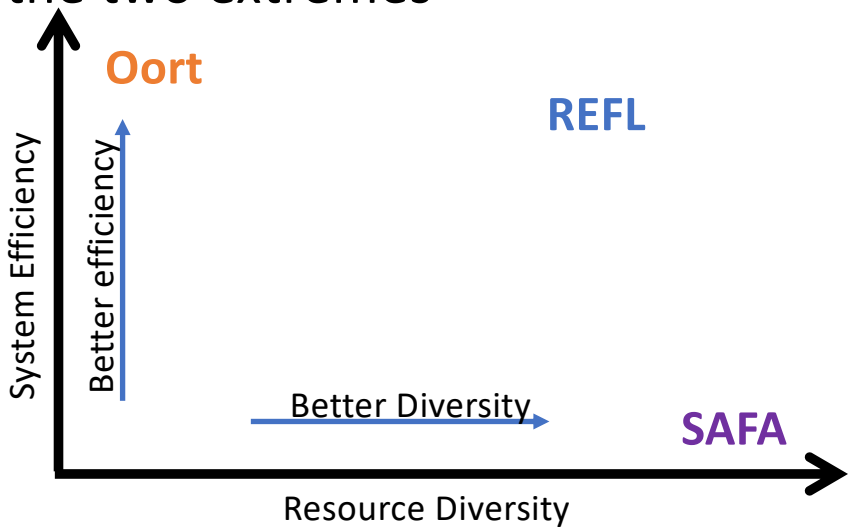
Motivation – Resource Efficiency

- We identify a Trade-off between System Efficiency vs Resource Diversity
  - Oort: favor fast learners over slow ones → high efficiency & low diversity
  - SAFA: select every learner → high diversity & low efficiency
  - Existing systems ignore resource consumption of the learners
- Our goal is to strike a balance between the two extremes

REFL: Resource Efficient FL System

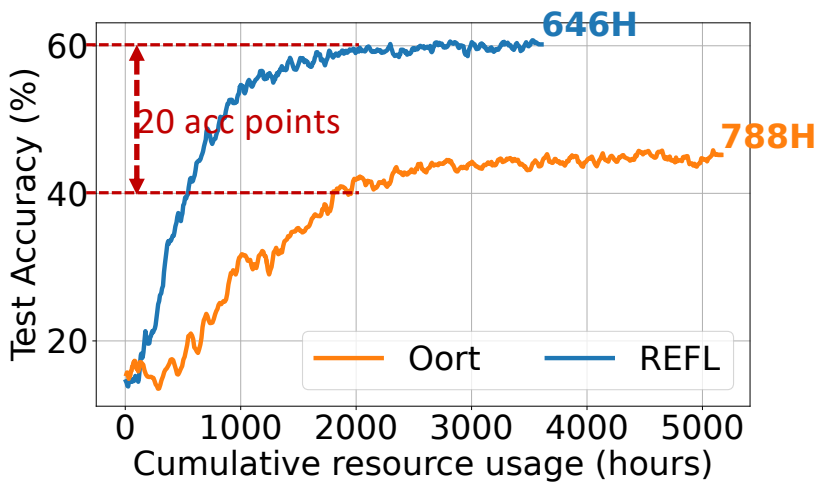
- ❑ Increases the **diversity** by prioritizing selection of least available learners
- ❑ Aggregates stale updates to boost the statistical **efficiency**
- ❑ Leads to reduced **resource** consumption

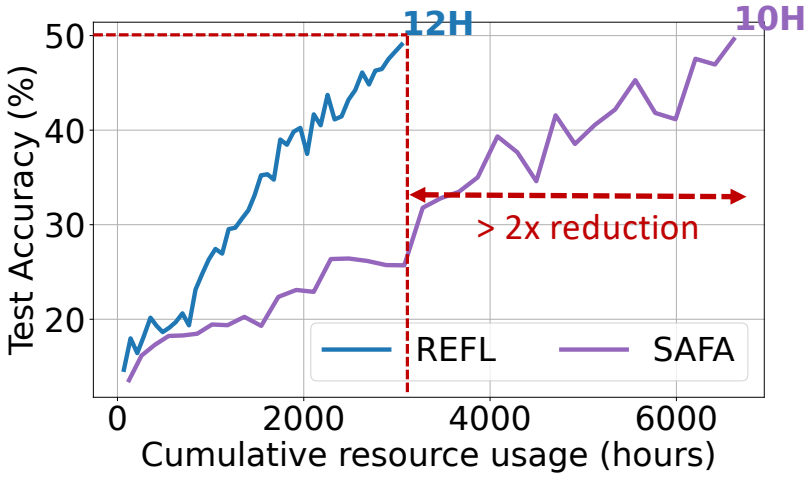
Plz check paper for details



Evaluation of REFL

- REFL → best model quality with least amount of resources and time
  - **Availability prioritization** leads to better diversity.
  - **Aggregation of stale updates** leads to resource savings.





EAFL+: Energy-Efficient Federated Learning

ACM SensSys 2024

Computer Communications, Elsevier 2024

https://ieeexplore.ieee.org/abstract/document/10562161

https://www.sciencedirect.com/science/article/pii/S0140366424001348

CODE: https://github.com/SAYED-Sys-Lab/EAFL

Challenges in FL - Continued!

- Once the end devices are selected, they will unconditionally take part in the FL process, which ignores their current conditions

  Remaining battery, Computational cost, Communication cost, …

- Federated Learning tasks need to be resilient to client dropouts (due to running out of battery):

  Dynamic resource availability, diverse computational resources….

Client dropout impacts model performance, resulting in wasted resources and unfairness.

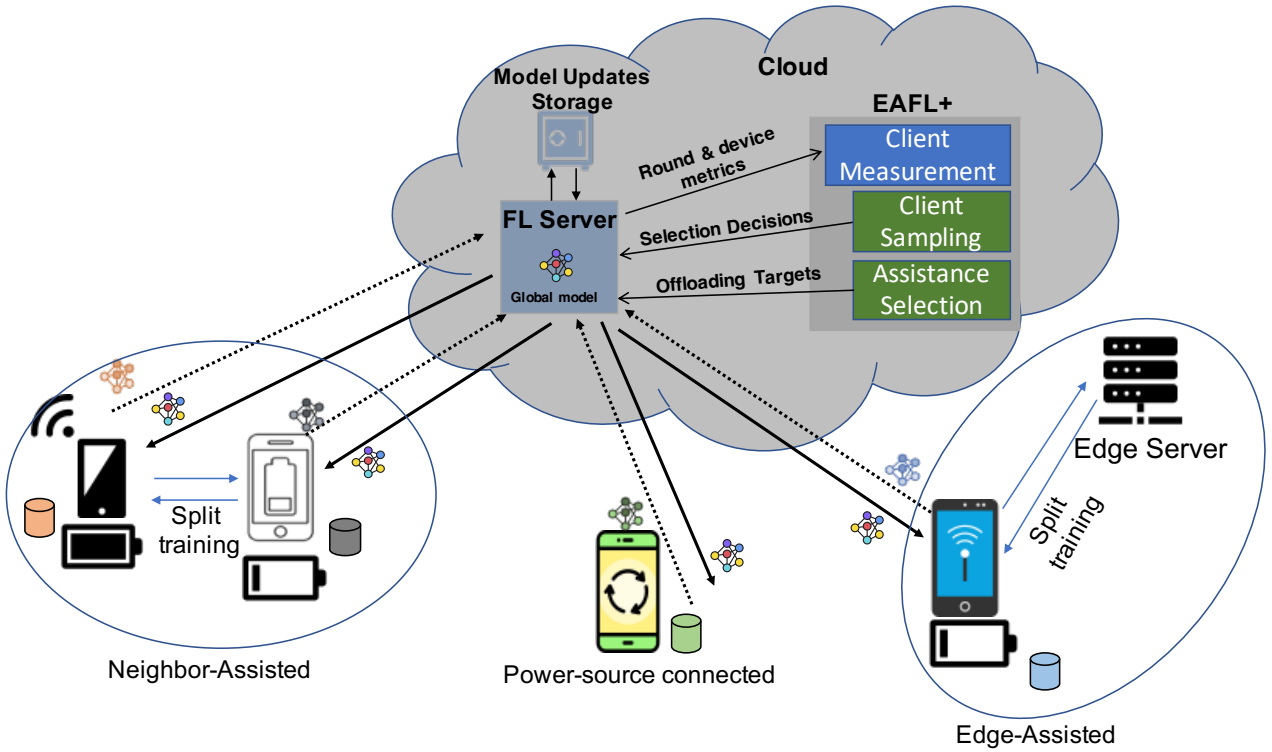
How to mitigate client dropouts in FL training in a Energy-heterogeneous setup?

Key Problems

- Heterogeneity of energy consumption characteristics of end devices.
- Lack of modeling or empirical/measurement research on real/prototype FL systems.
  - We conducted an empirical study on real mobile devices in our recent work
  - J. A. Esquivel, B. Aldous, A. M. Abdelmoniem, et al., "Performance Profiling of Federated Learning Across Heterogeneous Mobile Devices," *IEEE 24th International Conference on Software Quality, Reliability, and Security (QRS)*

EAFL+ System Overview

- Energy-Efficient FL requires:
  - Leveraging **task offloading between Edge-Cloud Continuum (in the form of split learning)** to reduce energy usage and ensure wider participation



EAFL+: Assist Scoring via Weighted Sum

Multi-metric/factor collection

$f_1, f_2, \ldots, f_n$

→ **High Level info (User/Admin) Input/Perf.**

→ **Estimation of relative importance vector**

$w_1, w_2, \ldots, w_n$

→ **weighted aggregation of individual scores**

Score = $w_1 f_1 + w_2 f_2 + \ldots + w_n f_n$

Weighted Sum Scoring

Pseudocode

Algorithm 1: Assistance Target Selection

T **Input** : S- sampled clients , C_W- weak clients, $layer_c$- cut layer, T_{comp}- weak client comp time, T_{comm}- weak client comm time, T_{compt}- comp time of offloading point, T_{commt}- comm time of offloading point

Ouput: $target$-selected offloading point

SelectTargetpoint($S, C_w, layer_c$)

$attributes = 3$;

$neighbours \leftarrow AssociatedNodes\ (C_W)$;

$P = [Edge, Cloud, neighbours]$;

$Y = [[0 \text{ in len}(attributes)][0 \text{ in len}(P)]]$;

for n *in* P **do**

  for j *in range(attributes)* **do**

    $Y[n][j] = Energy(T_{comp}, T_{comm}, layer_c)$;

    $Y[n][j] = Energy_{sp}(n, T_{compt}, T_{commt}, layer_c)$;

    $Y[n][j] = PowerClient(n)$;

    $scores[n] \leftarrow Score\_allocation(Y)$;

  end

end

$index \leftarrow max\_score\_index(scores)$;

$target = P[index]$;

return $target$

By controlling the weights in the weighted sum, we can move along the objective space from one solution to the next:

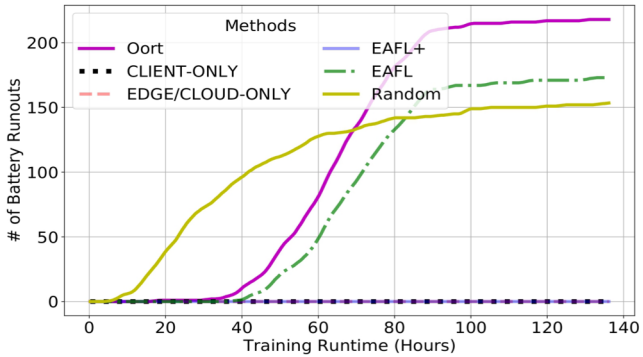
In this work, we assume either:

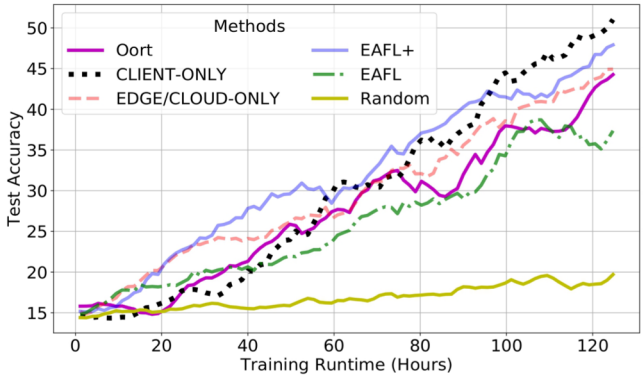
1) The weights to be equal for each metric/factor (but can be varied)
2) The weights are known or given (but can be estimated)

Check out the paper for more details

Evaluation: Results

- We compare EAFL+ vs selection methods with no offloading:
  - **Random** as Baseline (select clients randomly, no offload)
  - **Oort** (intelligent client selection to optimise efficiency)
  - **EAFL** (intelligent client selection to optimise efficiency & energy)
  - **EAFL+** and its variants (Client-Only vs Edge/Cloud-Only offload)
- The results (top-right) clearly show the benefits of different EAFL+ variants in achieving the best accuracy.
  - More than +24% over EAFL and +9% over Oort
- The results (top-bottom) show that the EAFL+ can eliminate client dropouts and avoid model update loss.





- EAFL+ improves model accuracy by reducing battery runouts
  - Weak clients get to complete training and contribute to the model

CAD: Robust Federated Learning

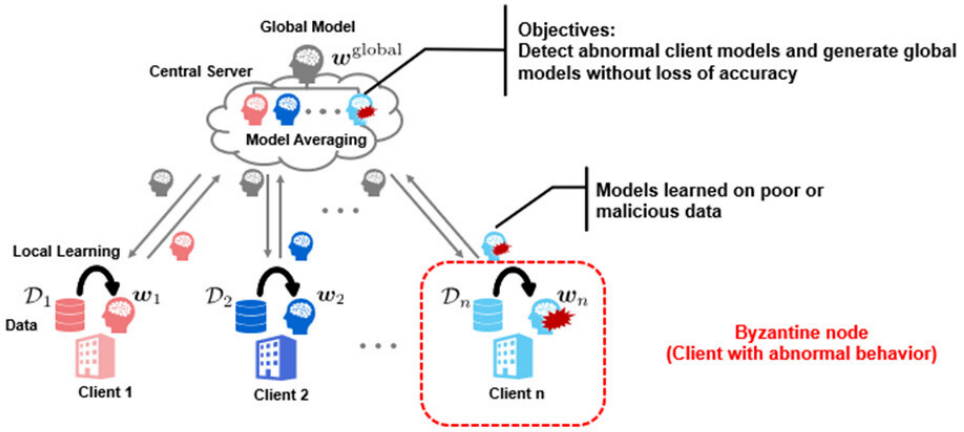
ArXiv 2024 / MobiUK workshop 2024

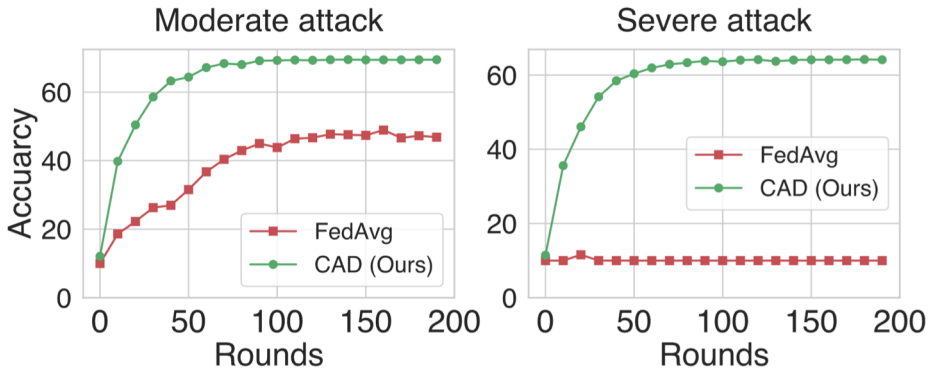
Under-review

https://arxiv.org/abs/2408.02813

Malicious or malformed clients

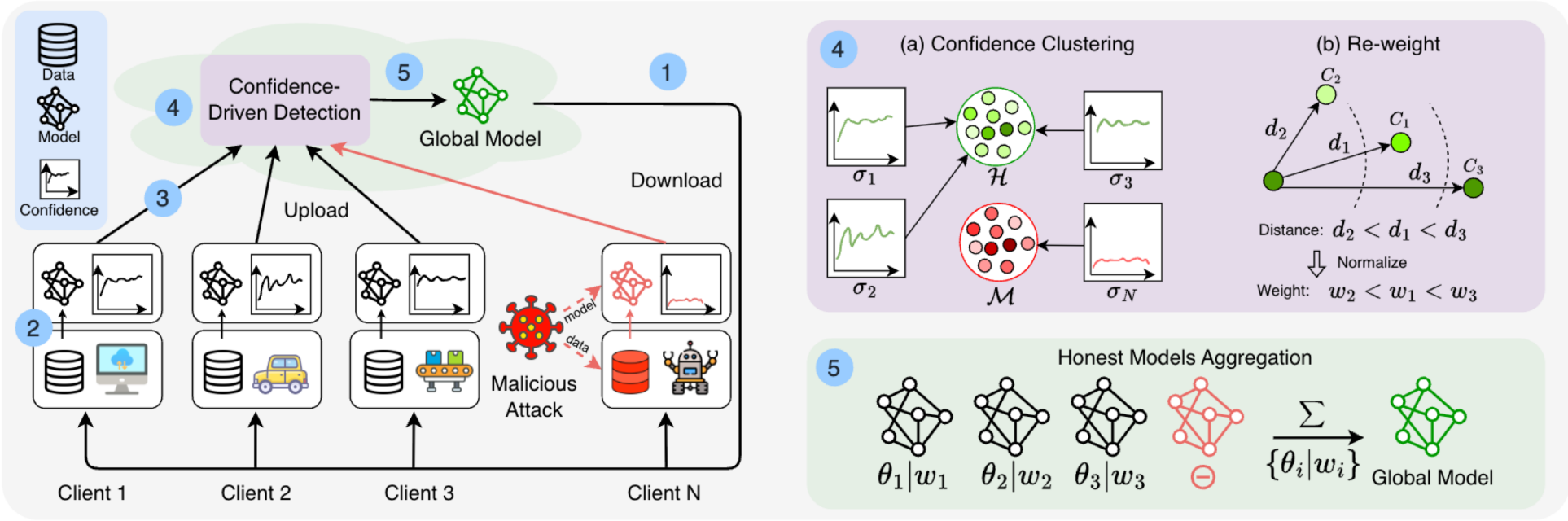
- Robust FL requires:
  - Identifying and mitigating malicious clients
  - Attacks include model and/or data poisoning





https://group.ntt/en/newsrelease/2024/05/07/240507a.html

CAD – Identify & Mitigate Malicious Clients



CAD – Robust to data/model attacks

Table 2: Robustness to various modes of attack on CIFAR-10 with VGG-11 model. Best results are shown in **bold**.

Intensity	Defence	Min-Max		Min-Sum		LIE		Label Shuffle	
		Full	Partial	Full	Partial	Full	Partial	Full	Partial
Moderate (25%)	FedAvg	55.40	64.37	49.44	59.20	49.84	52.89	68.32	68.60
	TrimMean	53.69	61.35	51.58	60.33	31.63	34.22	65.20	65.20
	AFA	68.89	68.71	68.99	69.17	69.09	68.80	68.88	69.09
	FLDetector	32.66	32.37	25.93	35.61	61.94	56.92	39.74	46.32
	DeFL	65.67	69.04	62.97	68.74	48.02	51.51	67.95	68.57
	FedRoLA	69.23	69.03	68.59	68.76	68.67	68.95	68.48	68.59
	CAD (Ours)	**69.85**	**69.43**	**69.94**	**69.43**	**69.67**	**69.18**	**69.36**	**69.47**
Severe (50%)	FedAvg	19.02	18.37	21.40	22.25	12.94	15.04	62.91	62.70
	TrimMean	16.92	17.27	17.49	17.20	11.55	12.35	37.26	36.88
	AFA	16.65	17.89	15.42	17.42	10.42	10.56	60.32	60.85
	FLDetector	23.61	20.02	20.47	20.73	14.41	11.96	34.84	38.81
	DeFL	21.60	27.50	48.26	48.26	23.17	23.95	63.02	62.31
	FedRoLA	19.52	19.60	21.40	22.10	13.42	14.17	62.52	61.98
	CAD (Ours)	**64.00**	**65.07**	**64.75**	**65.13**	**64.44**	**64.96**	**64.27**	**64.19**
Extreme (75%)	FedAvg	16.71	17.72	17.79	17.60	10.56	12.08	50.51	53.46
	TrimMean	14.57	15.03	17.52	16.60	10.01	10.42	21.17	20.23
	AFA	11.54	11.62	13.84	10.03	10.06	10.51	20.23	22.17
	FLDetector	16.71	17.72	17.79	16.31	11.04	12.08	18.33	18.04
	DeFL	20.11	19.44	19.42	20.02	16.06	16.76	52.18	52.72
	FedRoLA	16.71	16.86	17.79	17.59	10.00	10.00	54.39	52.46
	CAD (Ours)	**56.96**	**57.04**	**56.88**	**56.48**	**56.75**	**56.54**	**57.68**	**57.79**

Takeaways

- Heterogeneity is a significant challenge for FL:
  - Model quality degradations are not acceptable, esp. with diverging cases
  - Resource, energy and behaviour heterogeneity significantly impact the quality.

- To tackle heterogeneity → adapt to the system HW/Energy/Behaviour
  - **REFL** innovates on **client selection** and **stale updates aggregation** to yield a resource-efficient FL system → gains in quality, time and resource consumption
  - **EAFL+** leverages **offloading in the Edge-Cloud continuum** in battery-powered scenarios → , eliminating client dropouts, which boosts quality and fairness
  - **CAD** leverages **non-invasive confidence information** to ensure robustness → accurately identifies malicious clients and reduces their impact on global model

Hands-On Lab Time

Robust FL-based Intrusion Detection System

https://github.com/ahmedcs/Practical_FL_Tutorial/

Thanks

Q & A

To follow-up, please reach me at ahmed.sayed@qmul.ac.uk

If Interested in solving real-world problems!
Get in Touch!

For recent publication, see
https://scholar.google.com/citations?user=CzfuSJgAAAAJ&hl=en