EPFL

Exercise Session

Federated Learning

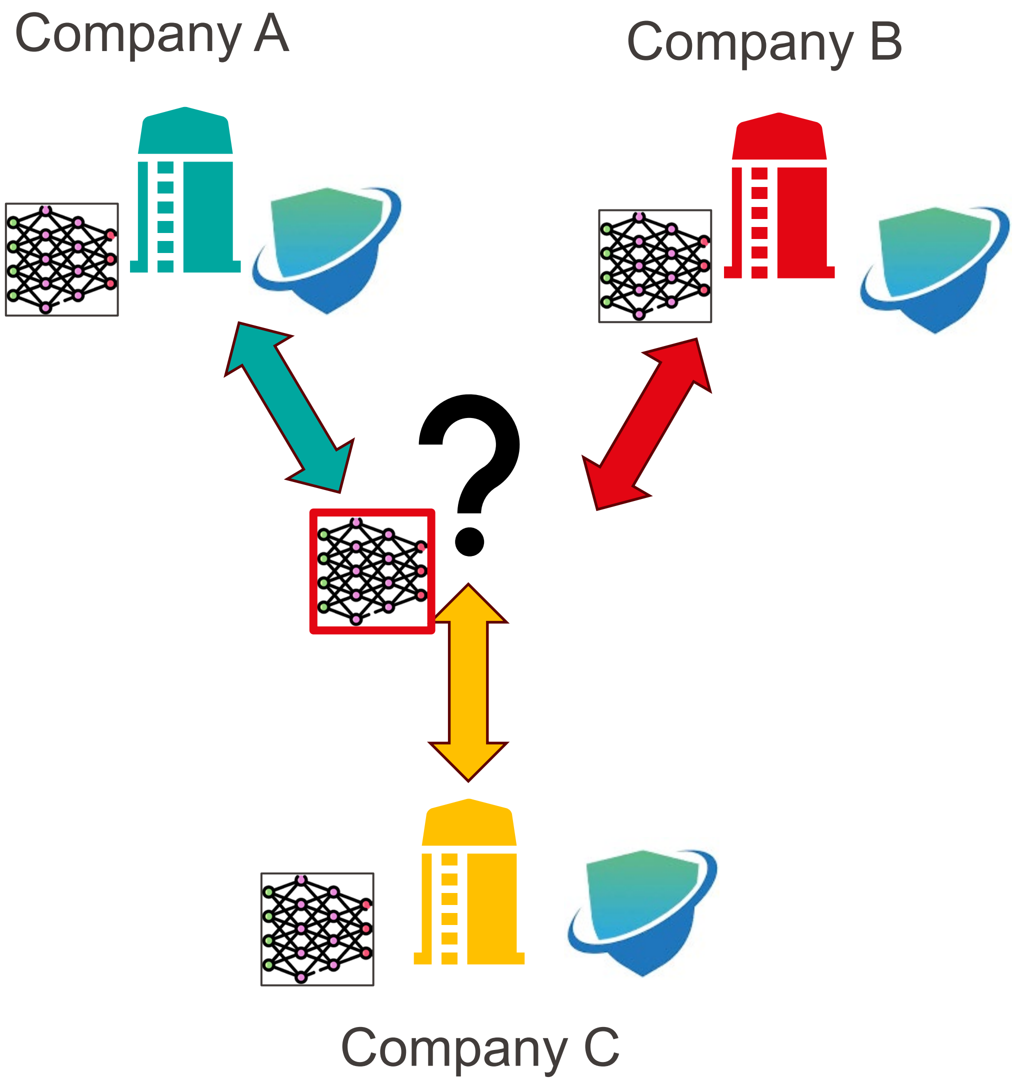
Leandro
21.11.2024

École polytechnique fédérale de Lausanne

Motivation

- Collaboration
- Privacy preservation

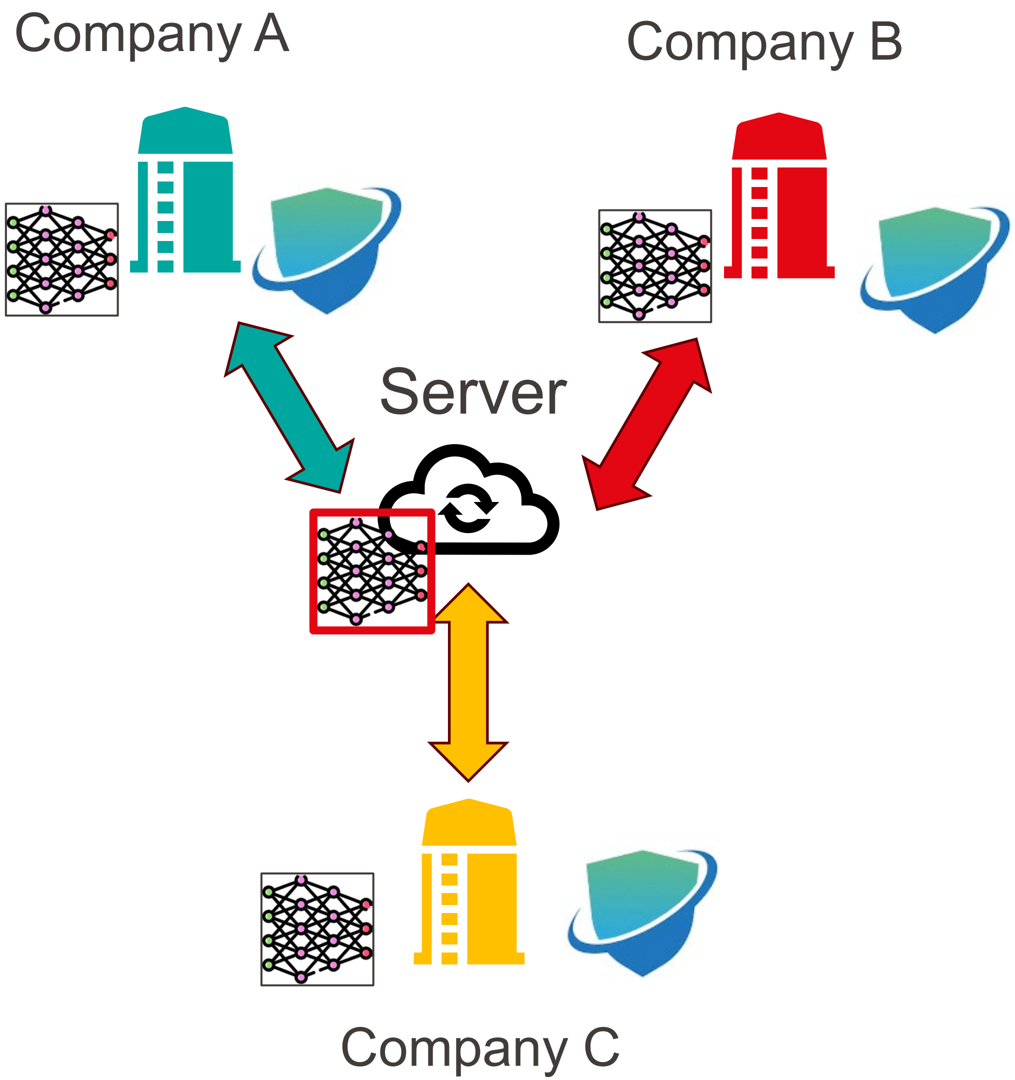
→ Federated Leaning
- More training data
- Privacy guaranteed



Motivation

- Collaboration
- Privacy preservation

➡ Federated Leaning
- More training data
- Privacy guaranteed



Global Perspective: Model vs. Weight Averaging

- Model averaging
  - Infrequent synchronization
  - Performance loss

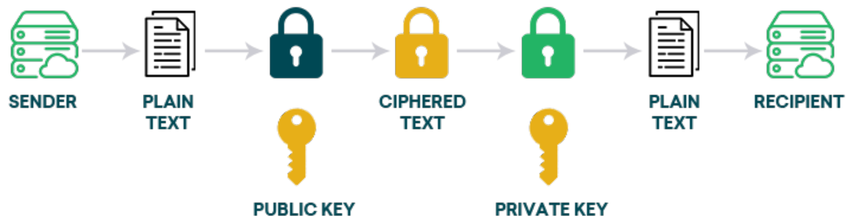
$$Model \propto \frac{1}{|I|}\sum_{i\in I} model_i$$

- Gradient averaging
  - Guaranteed convergence
  - Heavy communication

$$Update \propto \frac{1}{|I|}\sum_{i\in I} gradient_i$$

Privacy Concerns

- Homomorphic encryption
  - Rivest-Shamir-Adleman (RSA)
  - Prime factorization



https://www.clickssl.net/blog/what-is-rsa

- Differential privacy
  - $f = f_{basic} + Noise(Sensitivitvy_{model})$
  - Performance vs. privacy

Federated Learning: Set-up

- Collaborative Learning $f: X \rightarrow Y$

Horizontal Federation
Vertical Federation

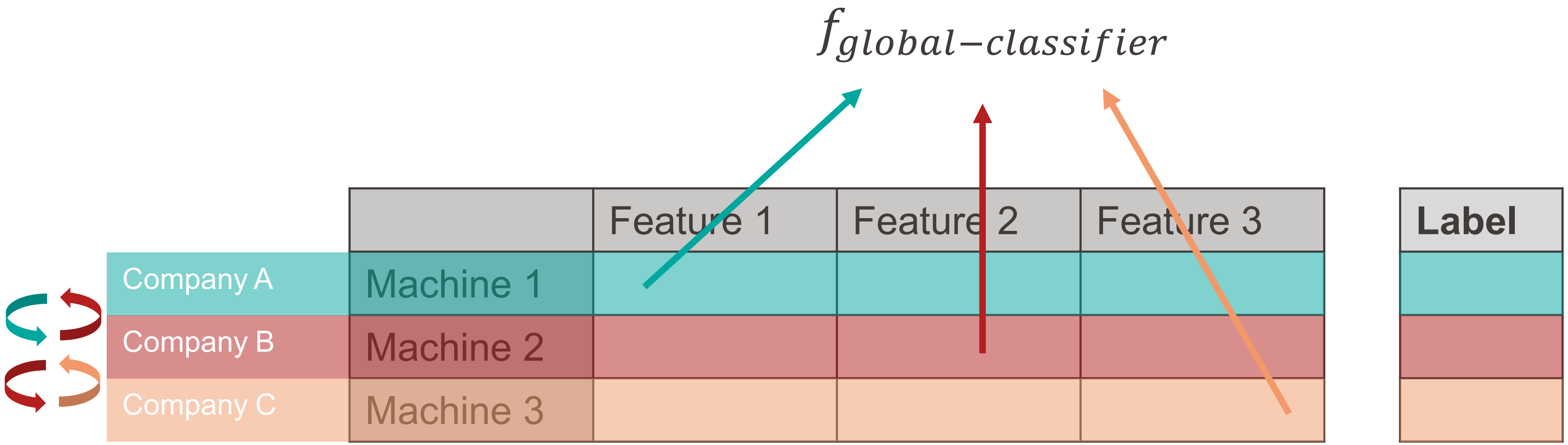
	Feature 1	Feature 2	Feature 3
Sample 1			
Sample 2			
Sample 3			

Label

Horizontal Federation

		Feature 1	Feature 2	Feature 3		Label
Company A	Machine 1					
Company B	Machine 2					
Company C	Machine 3					

Horizontal Federation



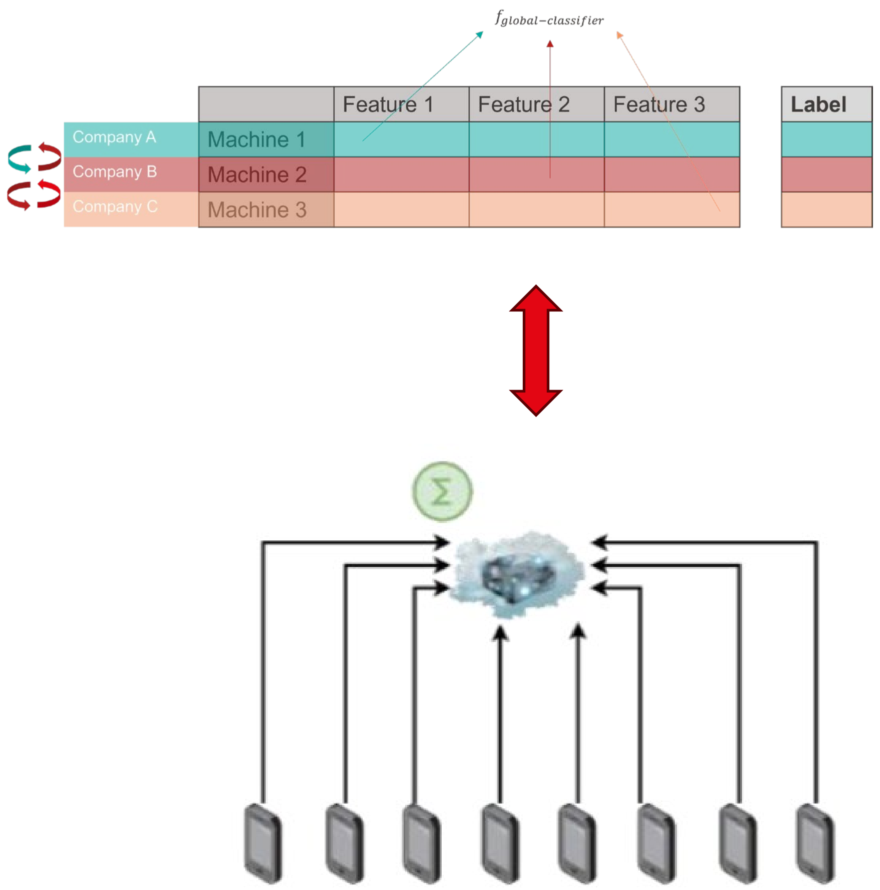
Horizontal Federation: Algorithm

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow \max(C \cdot K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate(k, w)**:** *// Run on client k*
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
  for each local epoch i from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return w to server

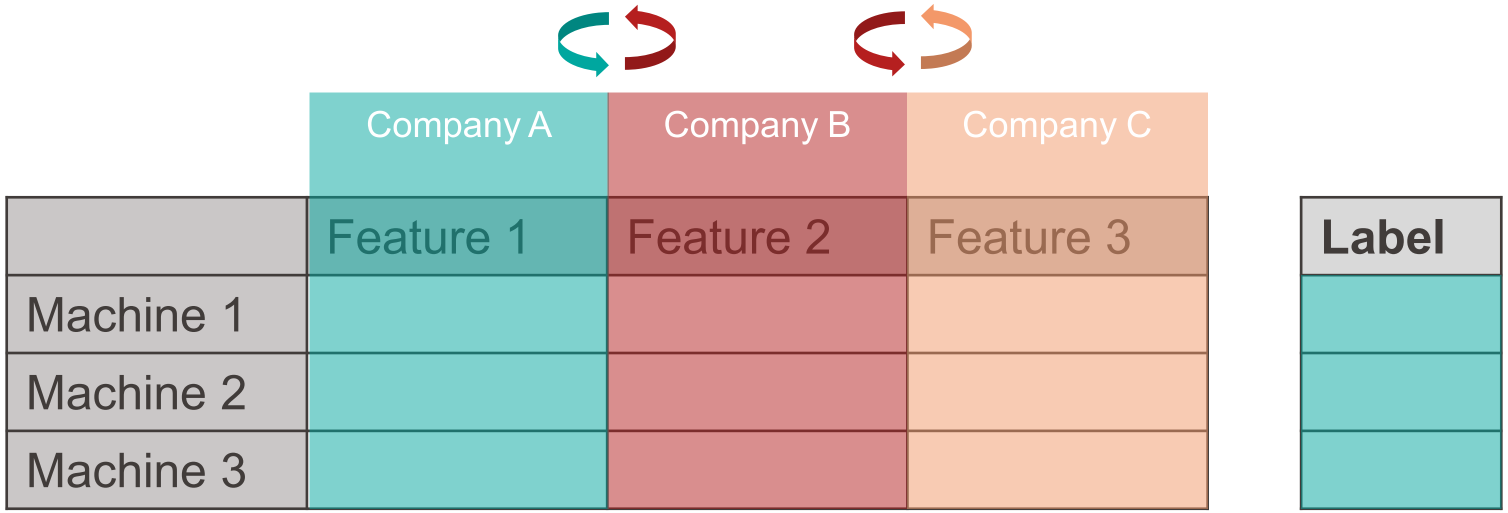
Communication-Efficient Learning of Deep Networks from Decentralized Data, McMahan B. et al



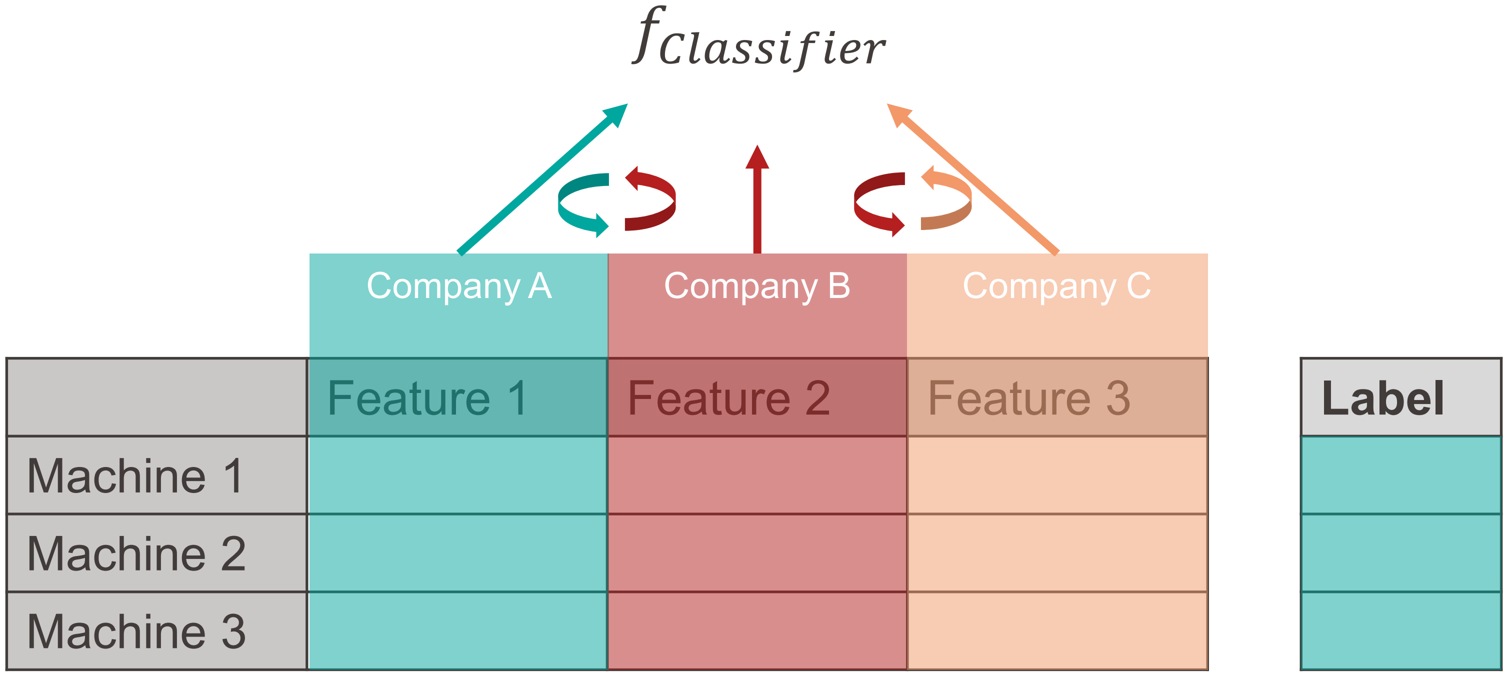
Horizontal Federation: Methods

- Neural Networks
- Gradient Boosting Regression Trees
- Random Forest

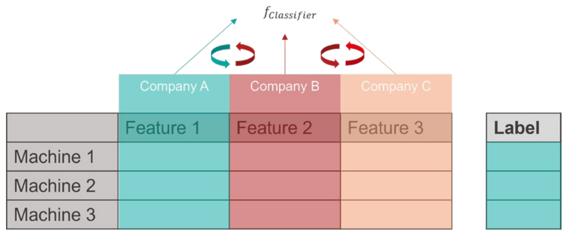
Vertical Federation



Vertical Federation



Vertical Federation: Algorithm



Algorithm for Linear Regression

$$[[\mathcal{L}]] = [[\mathcal{L}_A]] + [[\mathcal{L}_B]] + [[\mathcal{L}_{AB}]].$$

$$\min_{\Theta_A,\Theta_B} \sum_i \left\| \Theta_A x_i^A + \Theta_B x_i^B - y_i \right\|^2$$

$$\left[\left[\frac{\partial \mathcal{L}}{\partial \Theta_A}\right]\right] = 2\sum_i [[d_i]]\, x_i^A$$

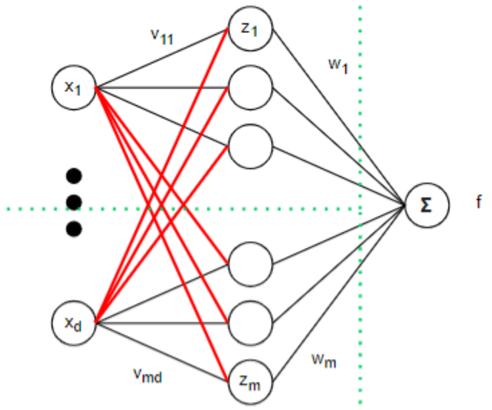
$$\left[\left[\frac{\partial \mathcal{L}}{\partial \Theta_B}\right]\right] = 2\sum_i [[d_i]]\, x_i^B .$$

	Party A	Party B	Party C
Step 1	Initializes Θ_A	Initializes Θ_B	Creates an encryption key pair and sends public key to A and B
Step 2	Computes $[[u_i^A]]$, $[[\mathcal{L}_A]]$ and sends to B	Computes $[[u_i^B]]$, $[[d_i^B]]$, $[[\mathcal{L}]]$, and sends $[[d_i^B]]$ to A, and sends $[[\mathcal{L}]]$ to C	
Step 3	Initializes R_A, computes $[[\frac{\partial \mathcal{L}}{\partial \Theta_A}]] + [[R_A]]$ and sends to C	Initializes R_B, computes $[[\frac{\partial \mathcal{L}}{\partial \Theta_B}]] + [[R_B]]$ and sends to C	Decrypts $[[\mathcal{L}]]$ and sends $[[\frac{\partial \mathcal{L}}{\partial \Theta_A}]] + R_A$ to A, $[[\frac{\partial \mathcal{L}}{\partial \Theta_B}]] + R_B$ to B
Step 4	Updates Θ_A	Updates Θ_B	
What is obtained?	Θ_A	Θ_B	

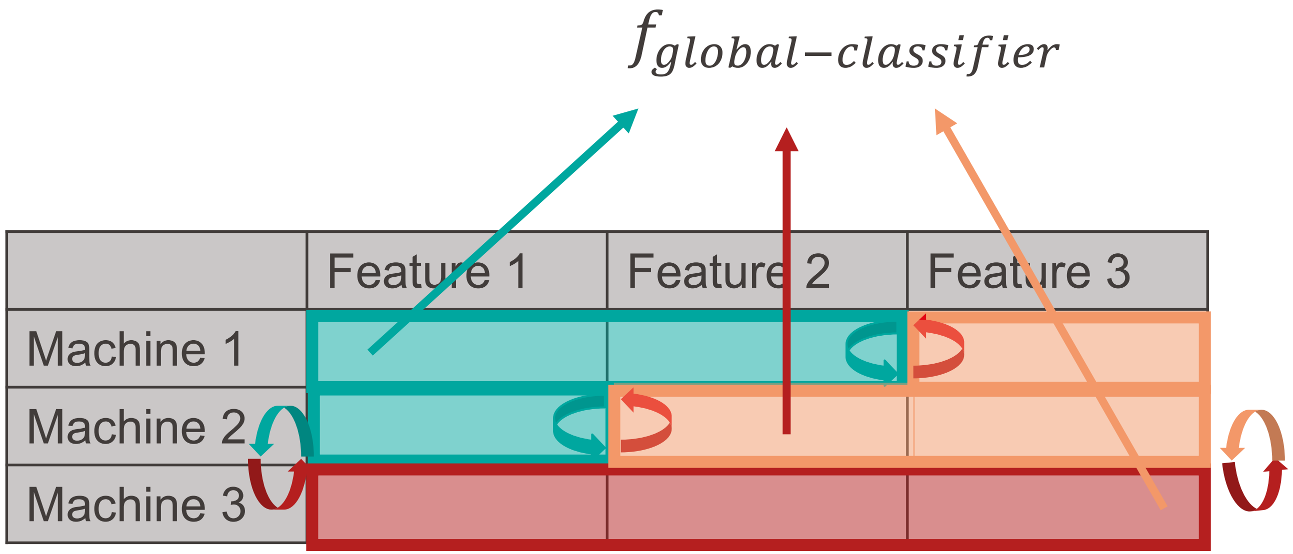
Federated Learning, Yang Q. et al.

Vertical Federation: Methods

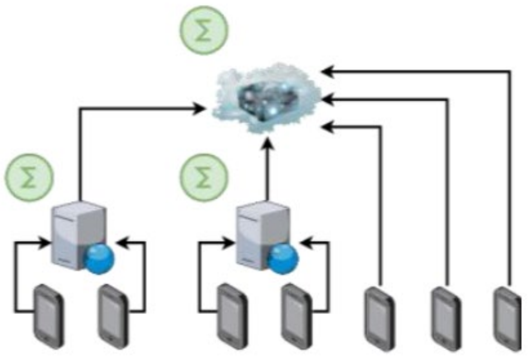
- Linear Regression
- Kernel Machines
- Neural Networks



Hybrid Federation



Hierarchical Federation



$$f_{global-classifier}$$

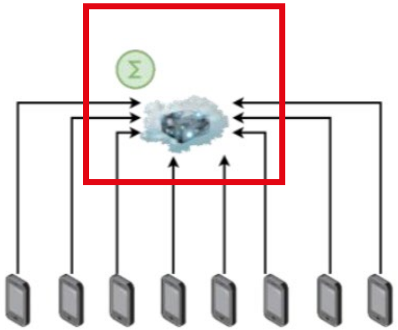
$$f_{AB-classifier}$$

		Feature 1	Feature 2	Feature 3	Label
Company A	Machine 1				
Company B	Machine 2				
Company C	Machine 3				

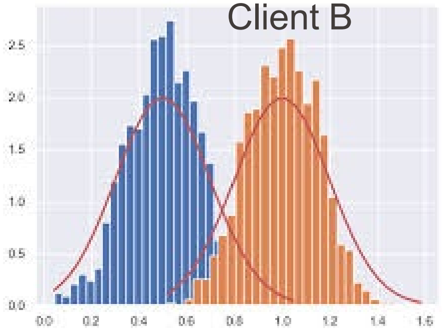
Federated Learning: Challenges

?

When to apply?



Server bottleneck



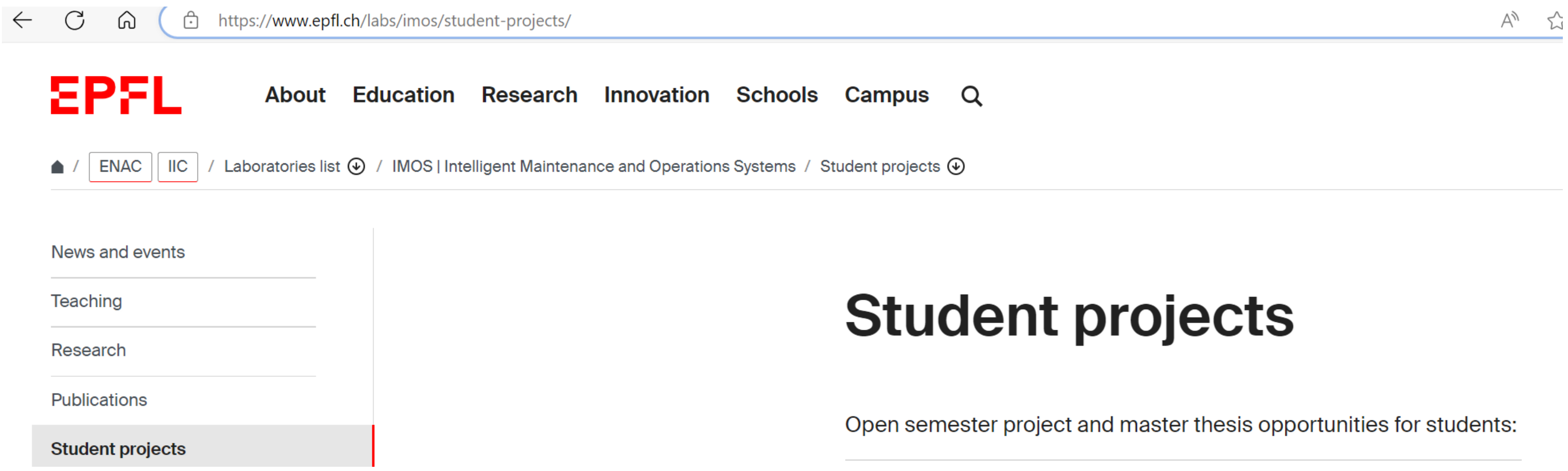
Distribution shift

Market design &
data valuation

Communication
frequency & loss

Privacy consideration
& unlearning

Student Projects



- Graph Neural Networks for Building Thermal Dynamics
- 3D Reconstrution with Thermal Imaging
- Ridig Object Dynamics from Videos

Thank You