Augmentation methods in Federated Learning

Dominik Lewy

Information Sensitivity: General\External

Agenda

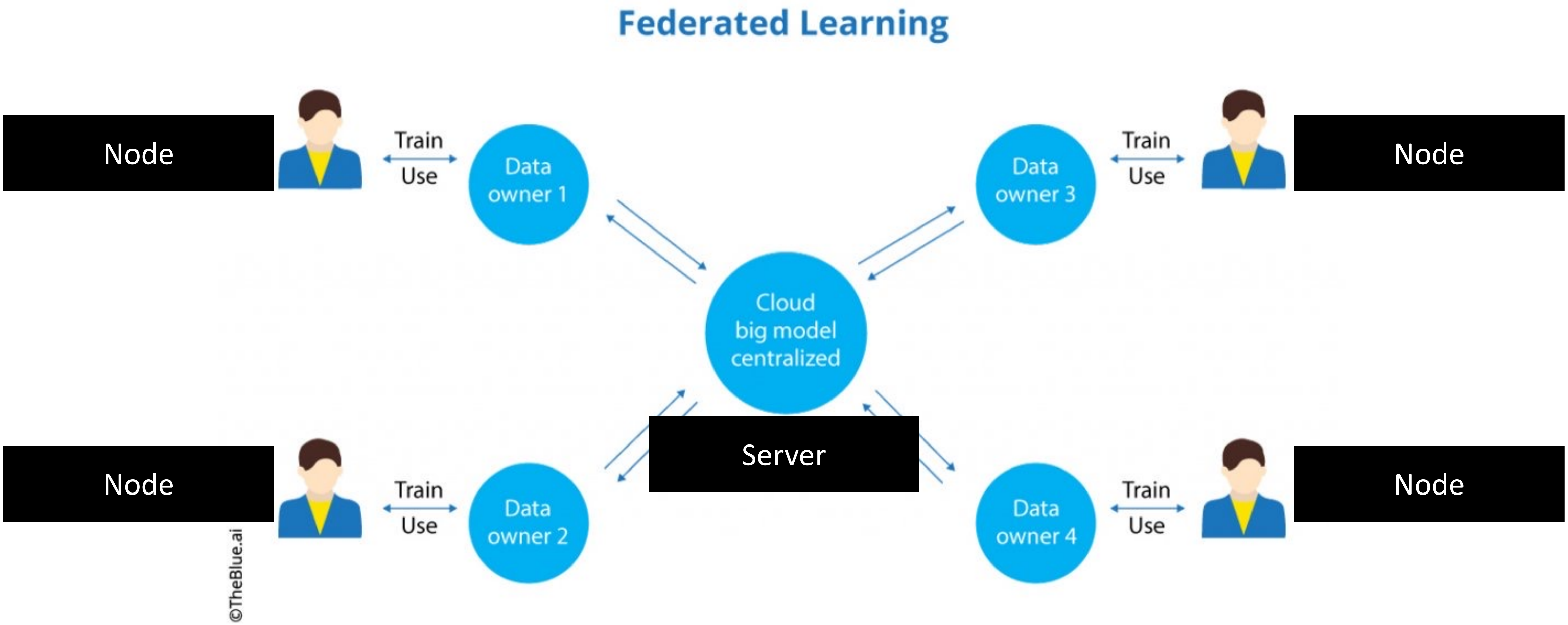
1. Federated Learning – introduction and general notation
2. Approaches to augmentation in Federated Learning
3. Federated Averaging (FedAvg) – canonical method in the space
4. Federated Mixup (FedMix)
5. StatMix – ICONIP 2022 – method presentation

Information Sensitivity: General\External

Federated Learning – introduction and general notation

Information Sensitivity: General\External

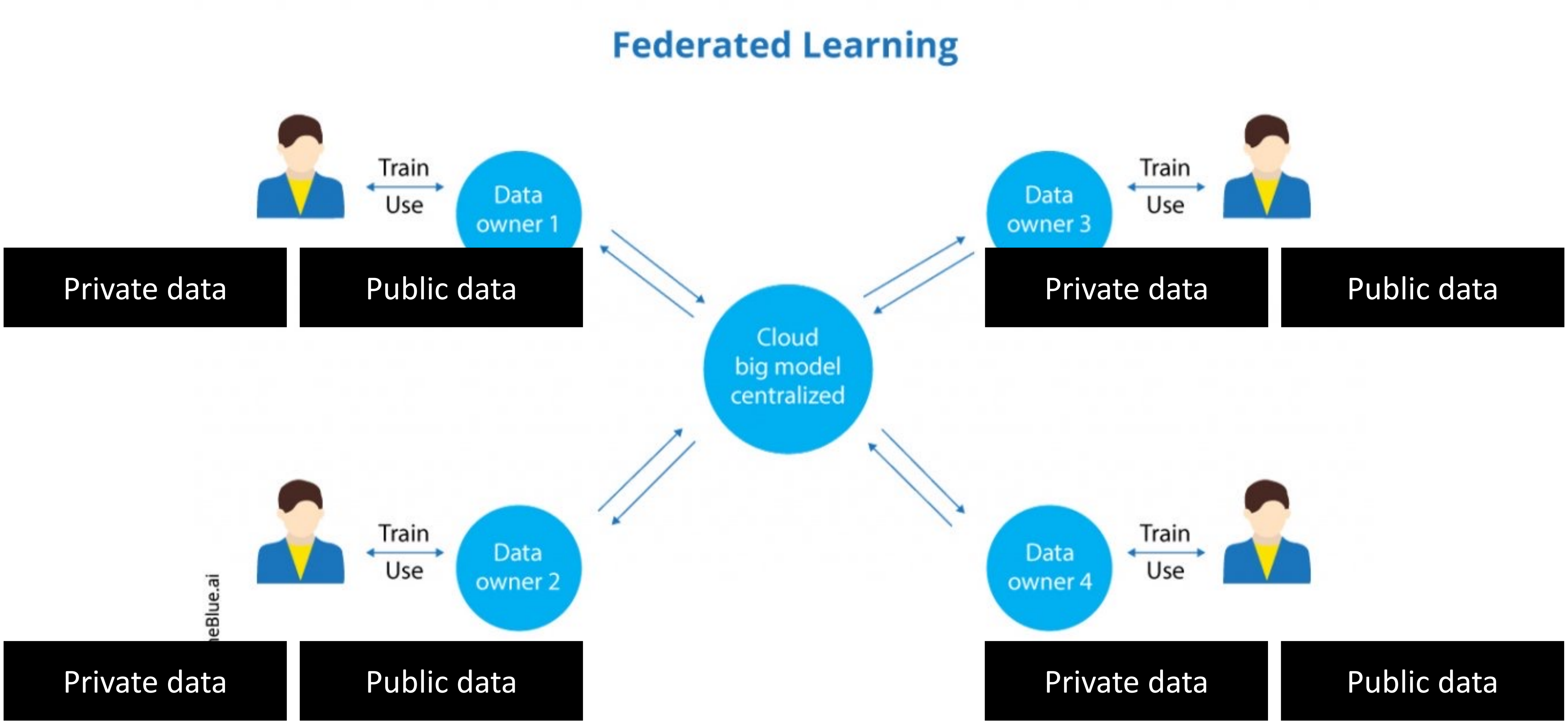
Federated Learning – introduction and general notation



Source: https://www.google.com/url?sa=i&url=https%3A%2F%2Ftheblue.ai%2Fblog-pl%2Ffederated-learning-2%2F&psig=AOvVaw2HJ0aM7PRIJV9lkdL-W9WX&ust=1665899766885000&source=images&cd=vfe&ved=0CA0QjRxqFwoTCMj5ibXG4foCFQAAAAAdAAAAABAI

Information Sensitivity: General\External

Federated Learning – introduction and general notation



Source: https://www.google.com/url?sa=i&url=https%3A%2F%2Ftheblue.ai%2Fblog-pl%2Ffederated-learning-2%2F&psig=AOvVaw2HJ0aM7PRIJV9lkdL-W9WX&ust=1665899766885000&source=images&cd=vfe&ved=0CA0QjRxqFwoTCMj5ibXG4foCFQAAAAAdAAAAABAI

Information Sensitivity: General\External

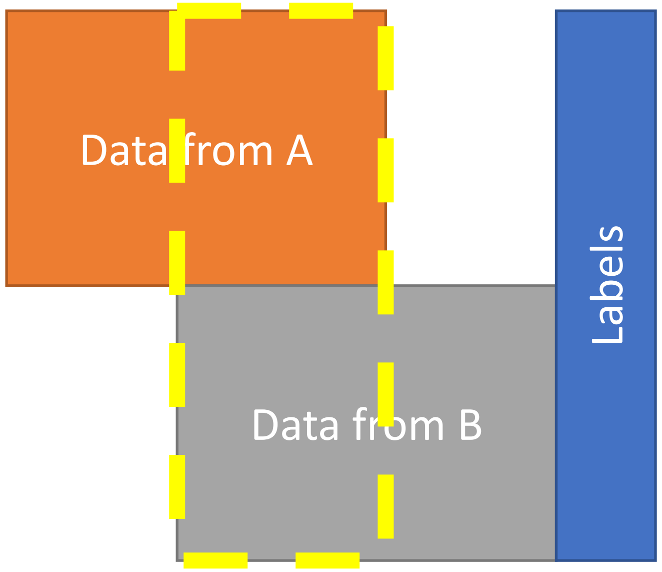
Taxonomy

Information Sensitivity: General\External

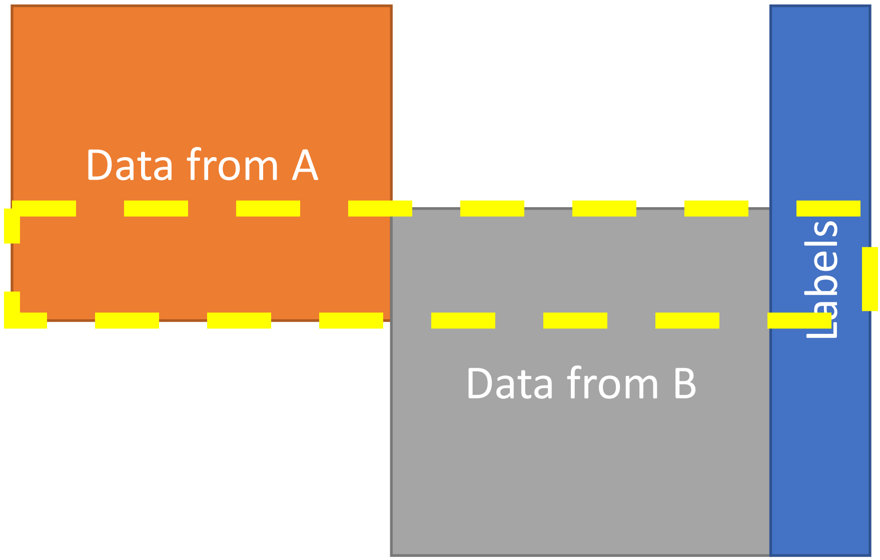
Source: https://arxiv.org/pdf/1907.09693.pdf

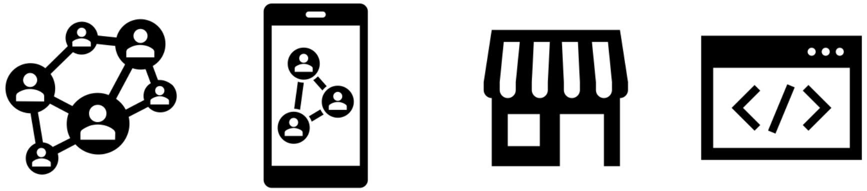
Taxonomy – Data partitioning

Horizontal – feature overlap, different observations:



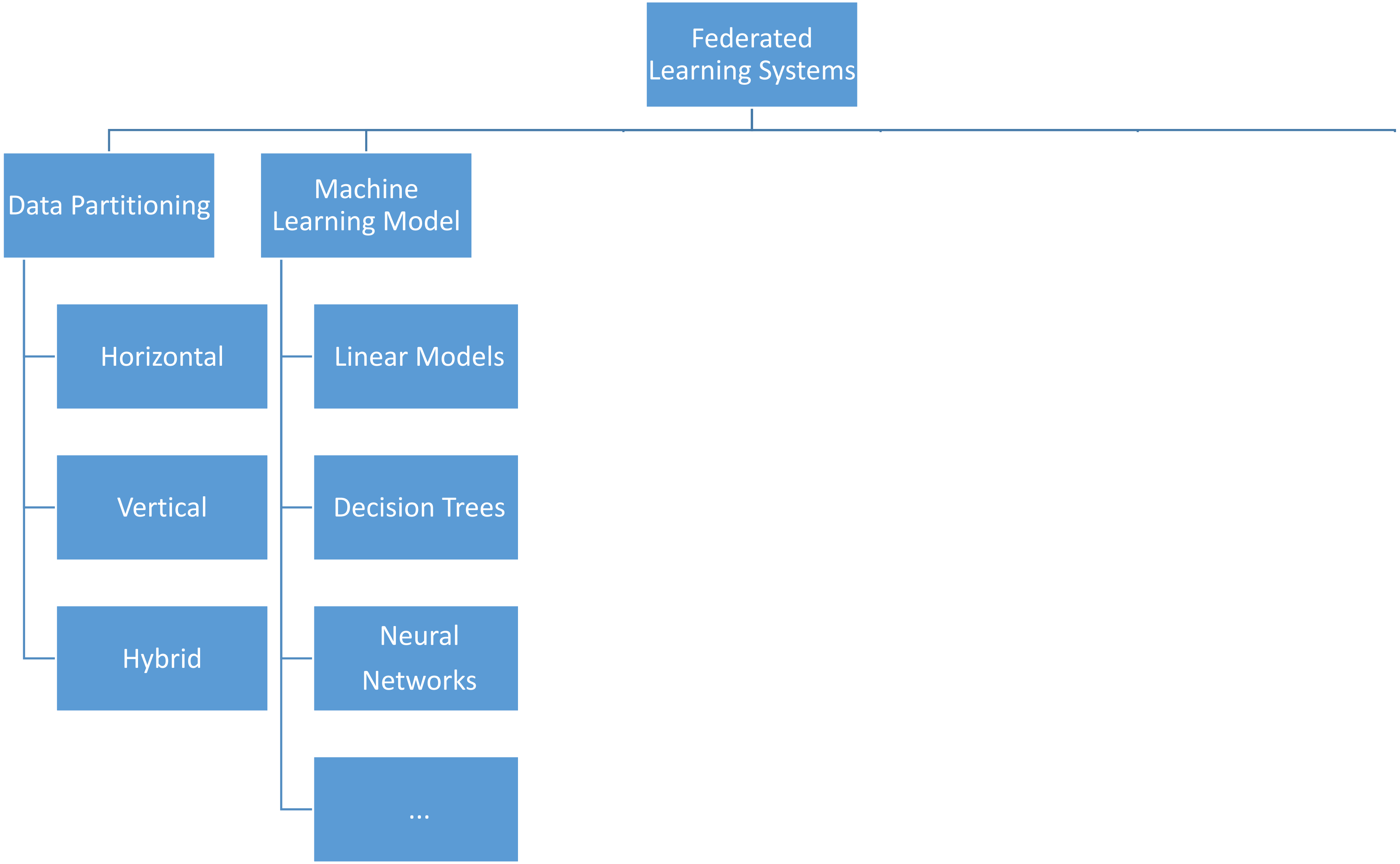
Vertical – different features, overlaping observations:





Source: own

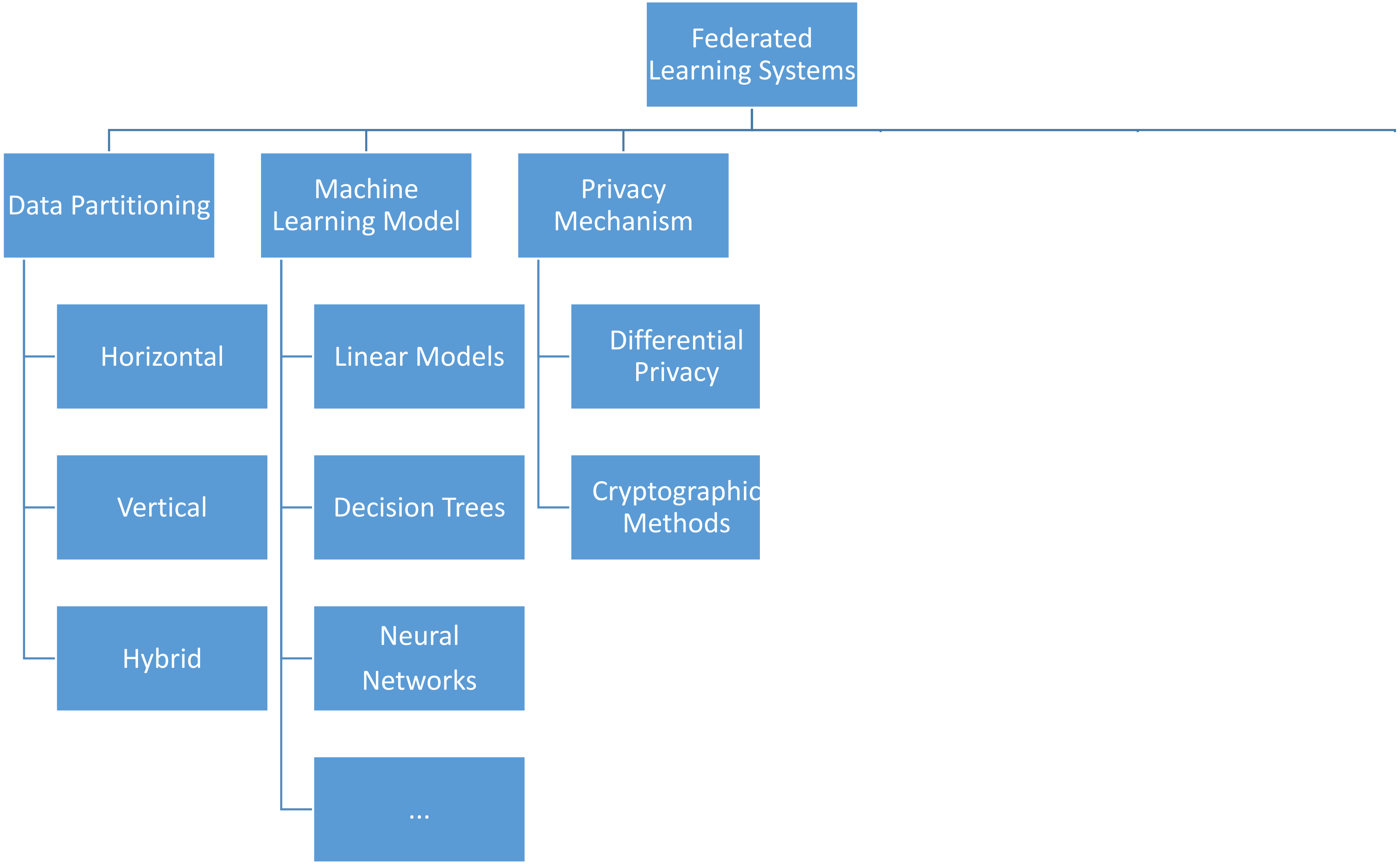
Taxonomy



Source: https://arxiv.org/pdf/1907.09693.pdf

Information Sensitivity: General\External

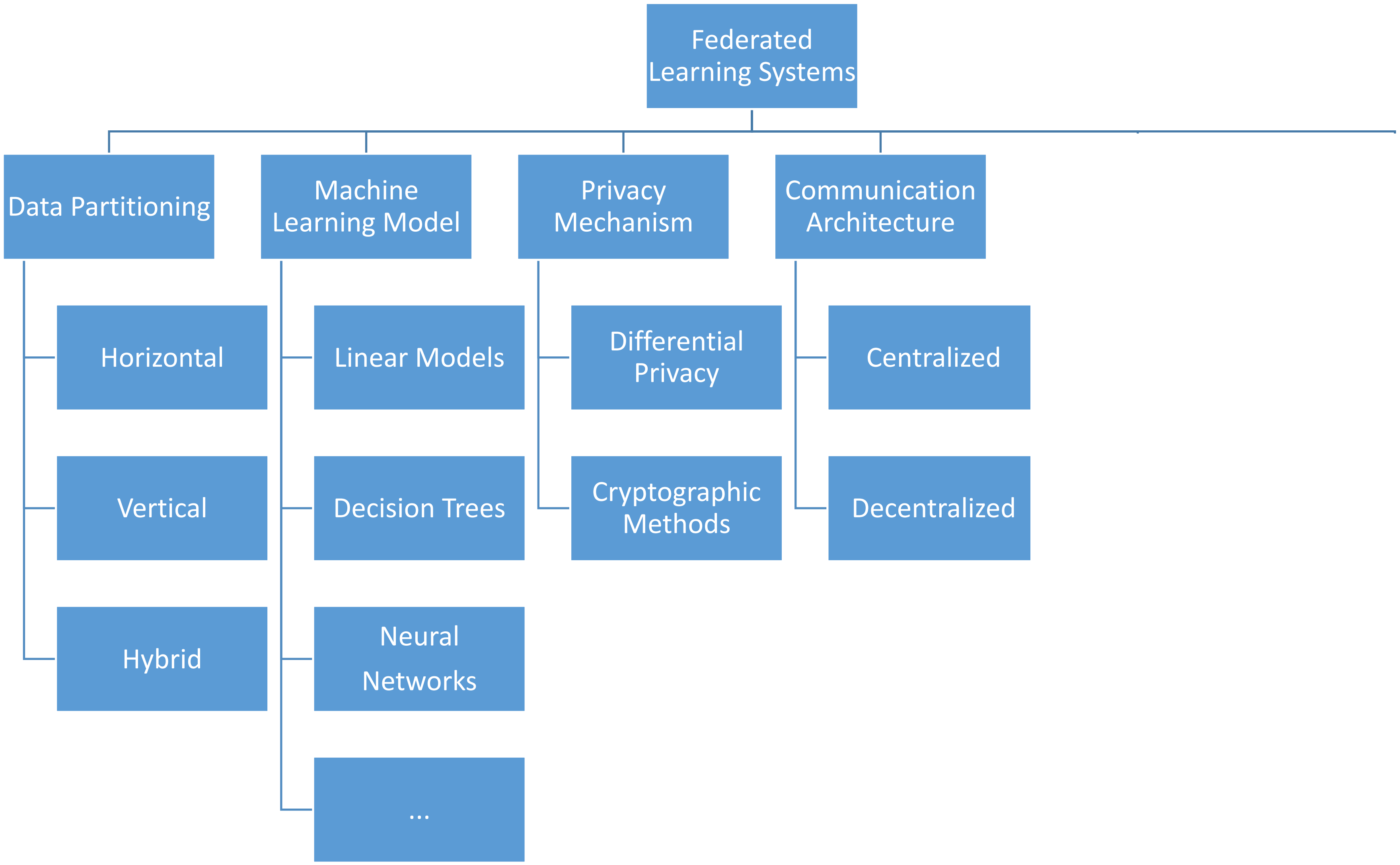
Taxonomy



Source: https://arxiv.org/pdf/1907.09693.pdf

Information Sensitivity: General\External

Taxonomy

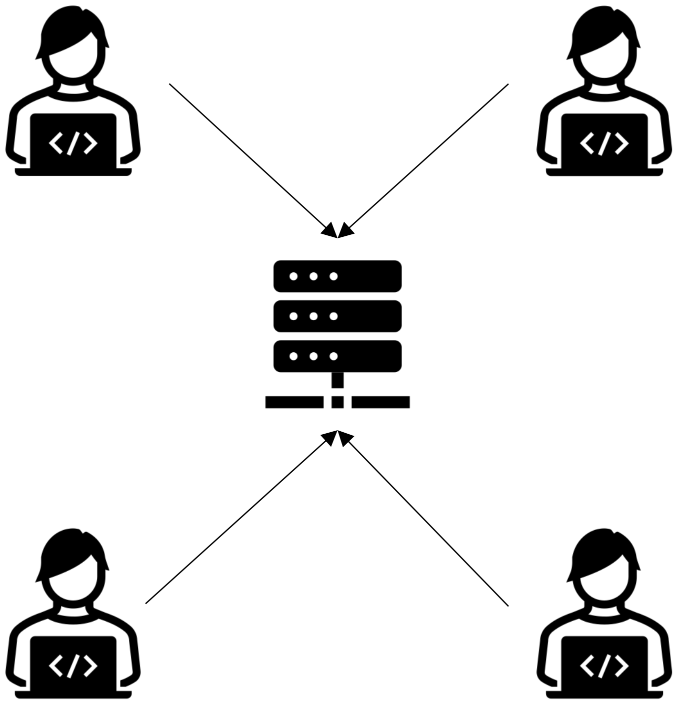


Source: https://arxiv.org/pdf/1907.09693.pdf

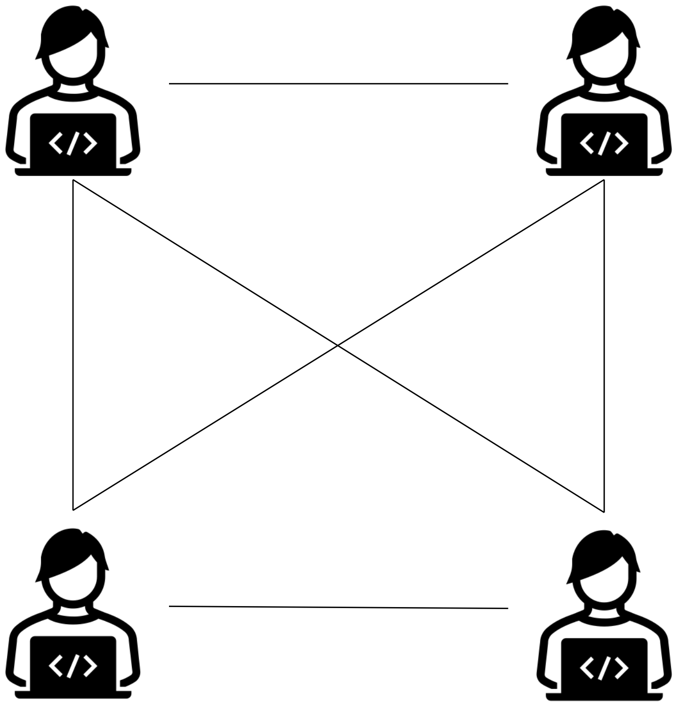
Information Sensitivity: General\External

Taxonomy – Communication Architecture

Server-orchestrated



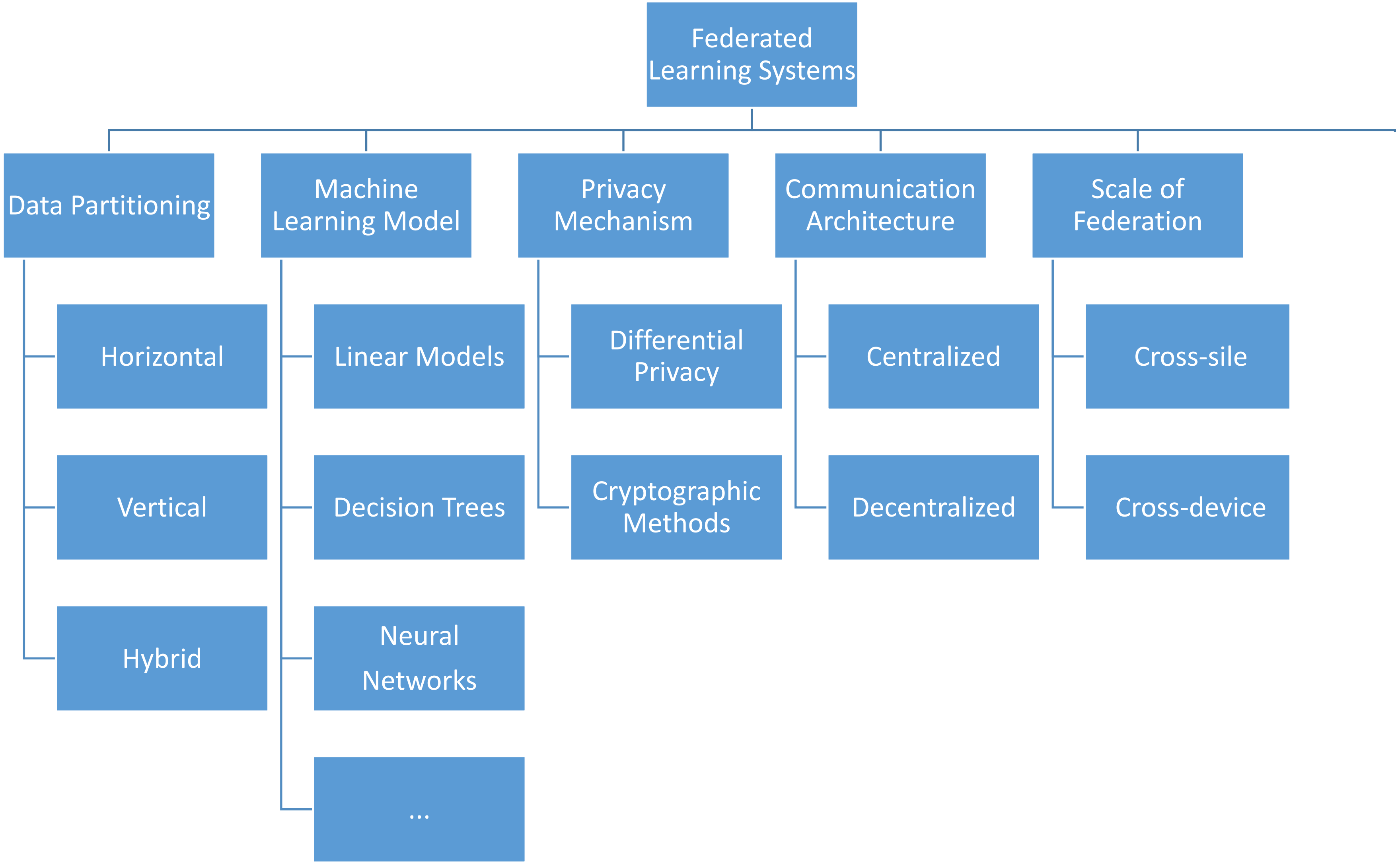
Fully decentralized



Source: own

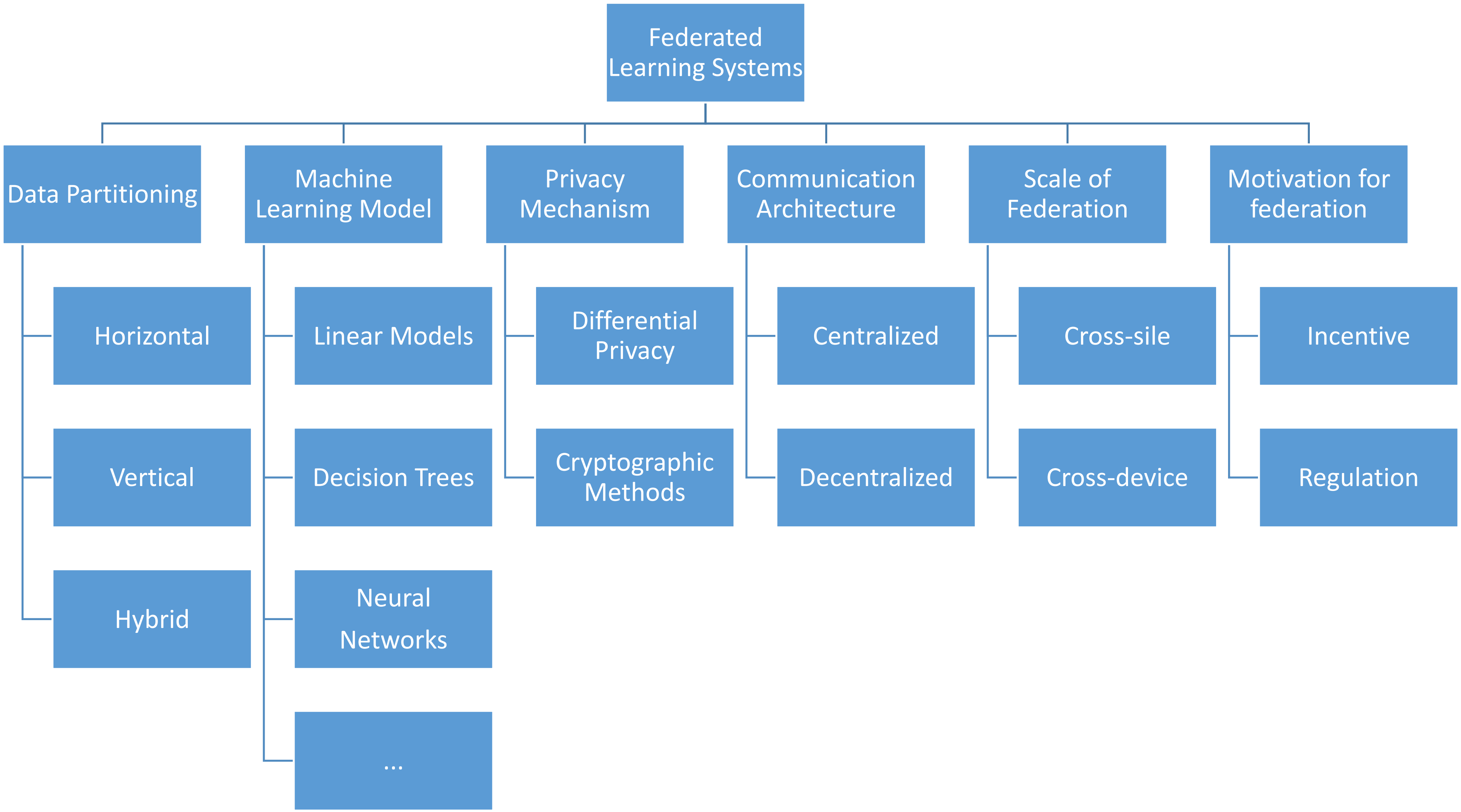
Information Sensitivity: General\External

Taxonomy



Source: https://arxiv.org/pdf/1907.09693.pdf

Taxonomy

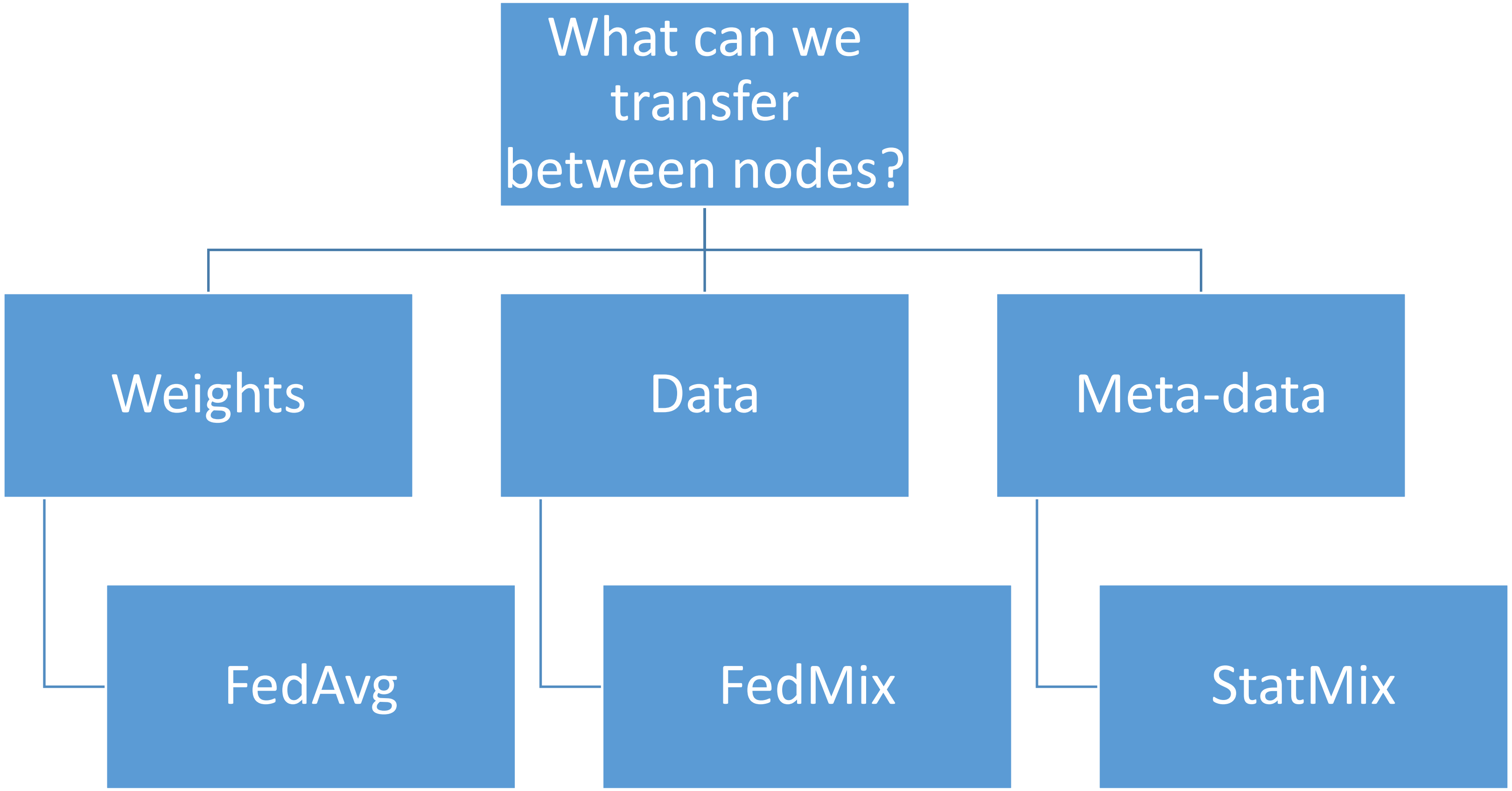


Source: https://arxiv.org/pdf/1907.09693.pdf

Approaches to augmentation in Federated Learning

Information Sensitivity: General\External

Approaches to augmentation in Federated Learning



Source: https://arxiv.org/pdf/1907.09693.pdf

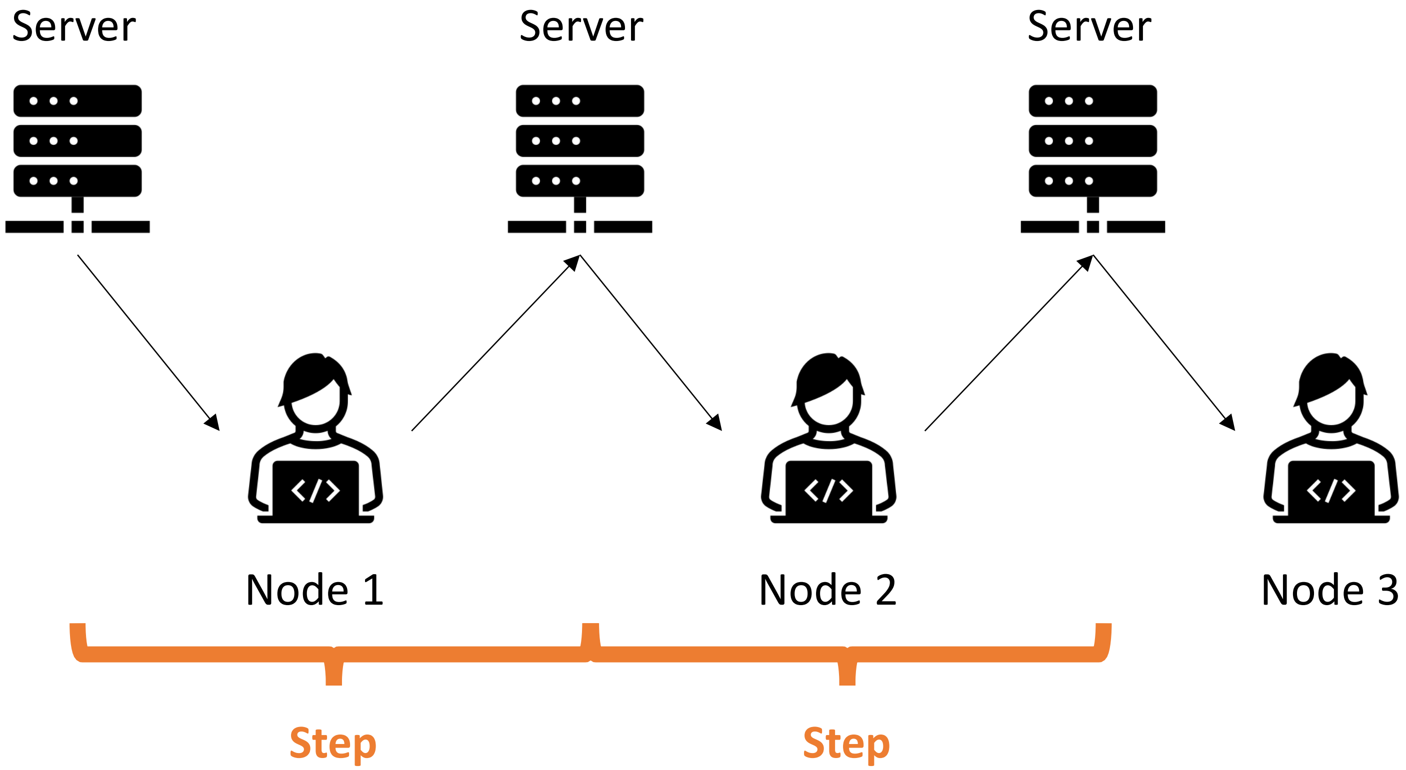
Information Sensitivity: General\External

Federated Averaging (FedAvg) – canonical method in the space

Information Sensitivity: General\External

FedAvg - motivation

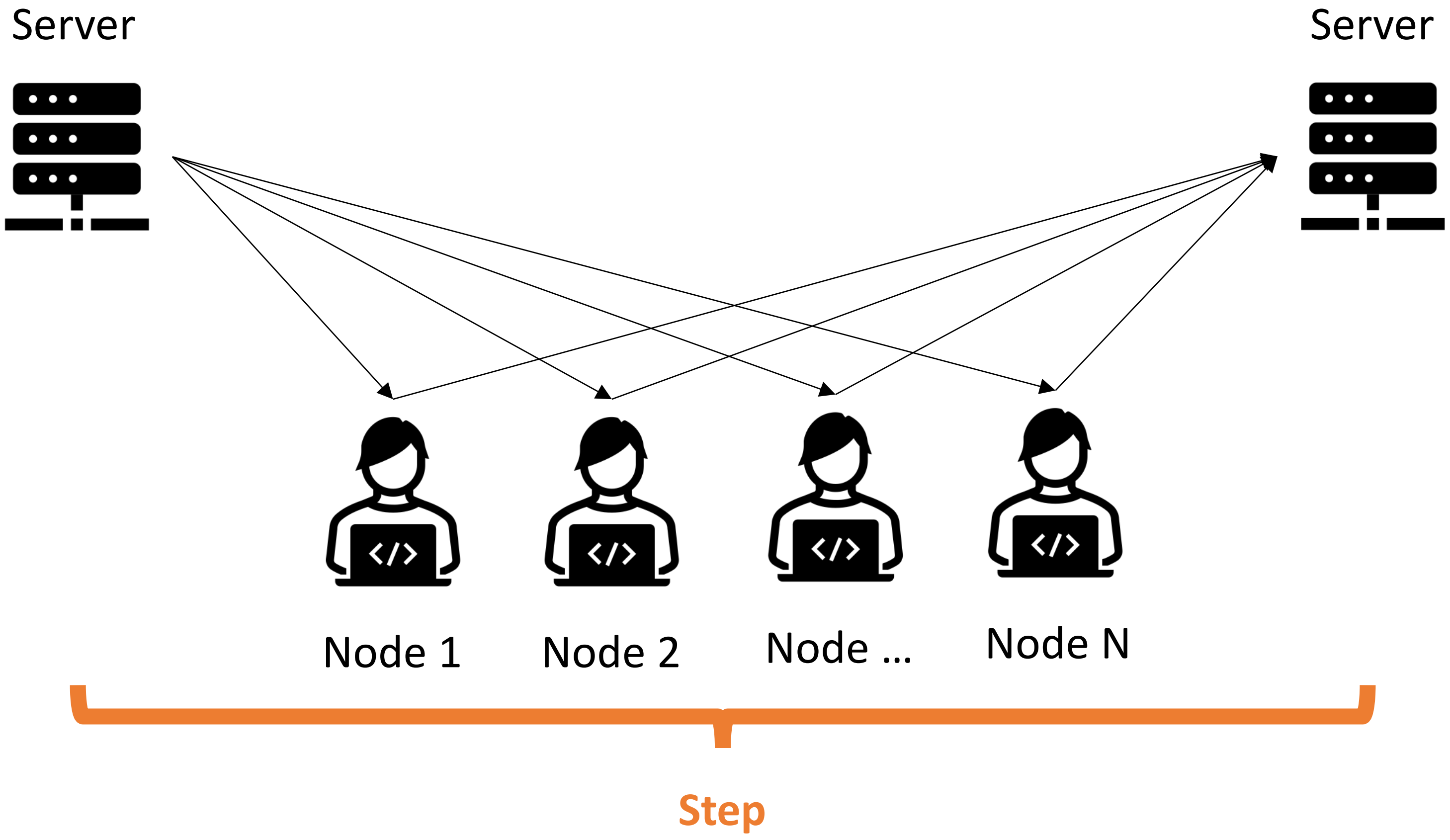
Naive application of SGD to federated optimization

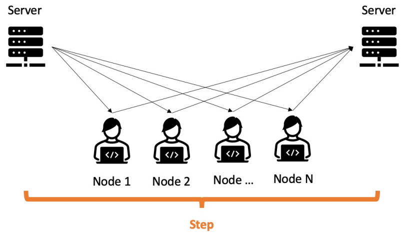


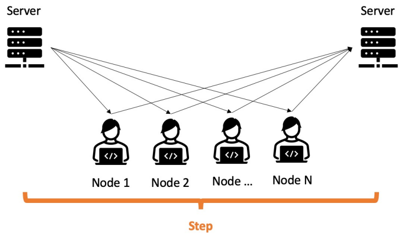
Source: https://arxiv.org/pdf/1602.05629.pdf

FedAvg - motivation

Naive application of SGD to federated optimization → FederatedSGD (FedSGD)





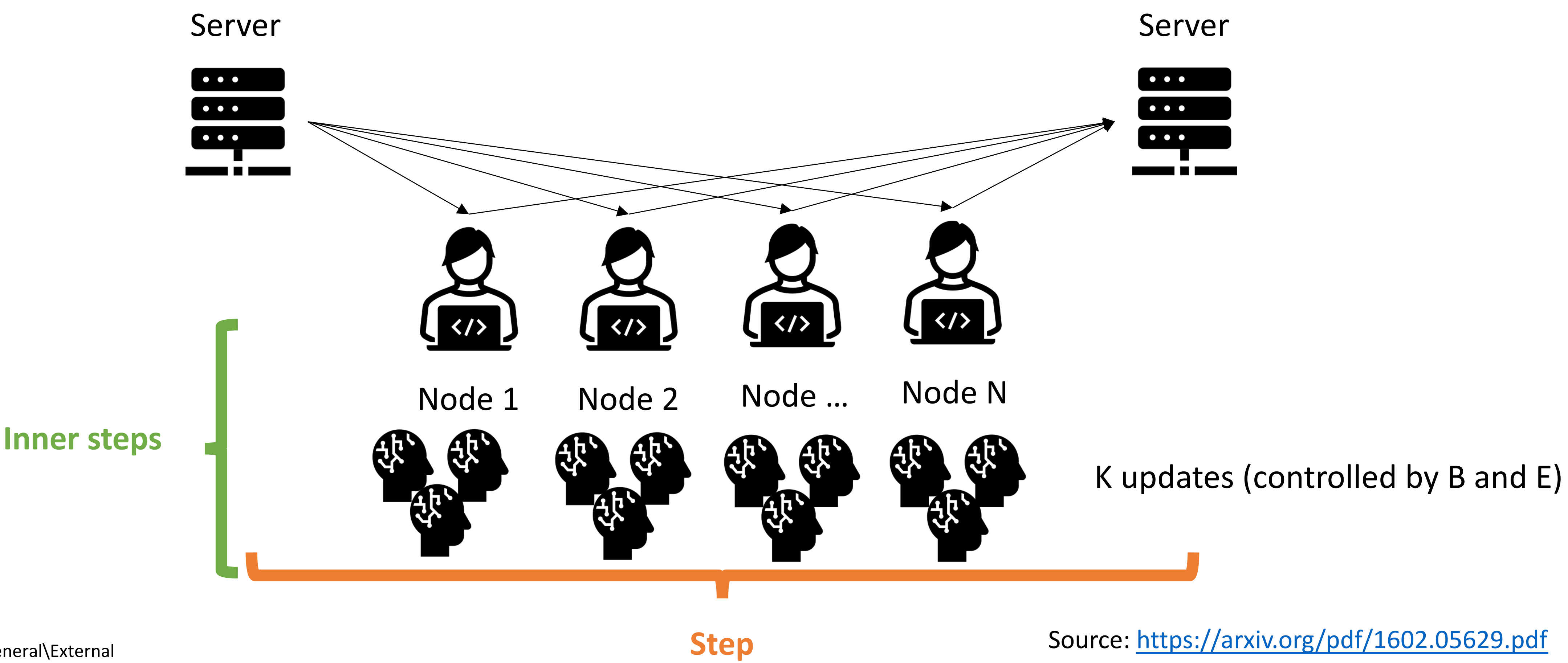


Source: https://arxiv.org/pdf/1602.05629.pdf

Information Sensitivity: General\External

FedAvg - motivation

Naive application of SGD to federated optimization → FederatedSGD (FedSGD) → FederatedAveraging (FedAvgg)



Source: https://arxiv.org/pdf/1602.05629.pdf

FedAvg - 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

Source: https://arxiv.org/pdf/1602.05629.pdf

FedAvg - results

Table 1: Effect of the client fraction C on the MNIST 2NN with $E = 1$ and CNN with $E = 5$. Note $C = 0.0$ corresponds to one client per round; since we use 100 clients for the MNIST data, the rows correspond to 1, 10 20, 50, and 100 clients. Each table entry gives the number of rounds of communication necessary to achieve a test-set accuracy of 97% for the 2NN and 99% for the CNN, along with the speedup relative to the $C = 0$ baseline. Five runs with the large batch size did not reach the target accuracy in the allowed time.

2NN	IID		Non-IID	
C	$B = \infty$	$B = 10$	$B = \infty$	$B = 10$
0.0	1455	316	4278	3275
0.1	1474 (1.0×)	87 (3.6×)	1796 (2.4×)	664 (4.9×)
0.2	1658 (0.9×)	77 (4.1×)	1528 (2.8×)	619 (5.3×)
0.5	— (—)	75 (4.2×)	— (—)	443 (7.4×)
1.0	— (—)	70 (4.5×)	— (—)	380 (8.6×)
CNN, $E = 5$				
0.0	387	50	1181	956
0.1	339 (1.1×)	18 (2.8×)	1100 (1.1×)	206 (4.6×)
0.2	337 (1.1×)	18 (2.8×)	978 (1.2×)	200 (4.8×)
0.5	164 (2.4×)	18 (2.8×)	1067 (1.1×)	261 (3.7×)
1.0	246 (1.6×)	16 (3.1×)	— (—)	97 (9.9×)

Source: https://arxiv.org/pdf/1602.05629.pdf

Information Sensitivity: General\External

FedAvg - results

C - % of nodes participating in comunication round
E – number of rounds epochs in each comunication round
B – mini-batch size

C=0.1

Table 2: Number of communication rounds to reach a target accuracy for FedAvg, versus FedSGD (first row, $E = 1$ and $B = \infty$). The u column gives $u = En/(KB)$, the expected number of updates per round.

MNIST CNN, 99% ACCURACY					
CNN	E	B	u	IID	NON-IID
FEDSGD	1	∞	1	626	483
FEDAVG	5	∞	5	179 (3.5×)	1000 (0.5×)
FEDAVG	1	50	12	65 (9.6×)	600 (0.8×)
FEDAVG	20	∞	20	234 (2.7×)	672 (0.7×)
FEDAVG	1	10	60	34 (18.4×)	350 (1.4×)
FEDAVG	5	50	60	29 (21.6×)	334 (1.4×)
FEDAVG	20	50	240	32 (19.6×)	426 (1.1×)
FEDAVG	5	10	300	20 (31.3×)	229 (2.1×)
FEDAVG	20	10	1200	18 (34.8×)	173 (2.8×)
SHAKESPEARE LSTM, 54% ACCURACY					
LSTM	E	B	u	IID	NON-IID
FEDSGD	1	∞	1.0	2488	3906
FEDAVG	1	50	1.5	1635 (1.5×)	549 (7.1×)
FEDAVG	5	∞	5.0	613 (4.1×)	597 (6.5×)
FEDAVG	1	10	7.4	460 (5.4×)	164 (23.8×)
FEDAVG	5	50	7.4	401 (6.2×)	152 (25.7×)
FEDAVG	5	10	37.1	192 (13.0×)	41 (95.3×)

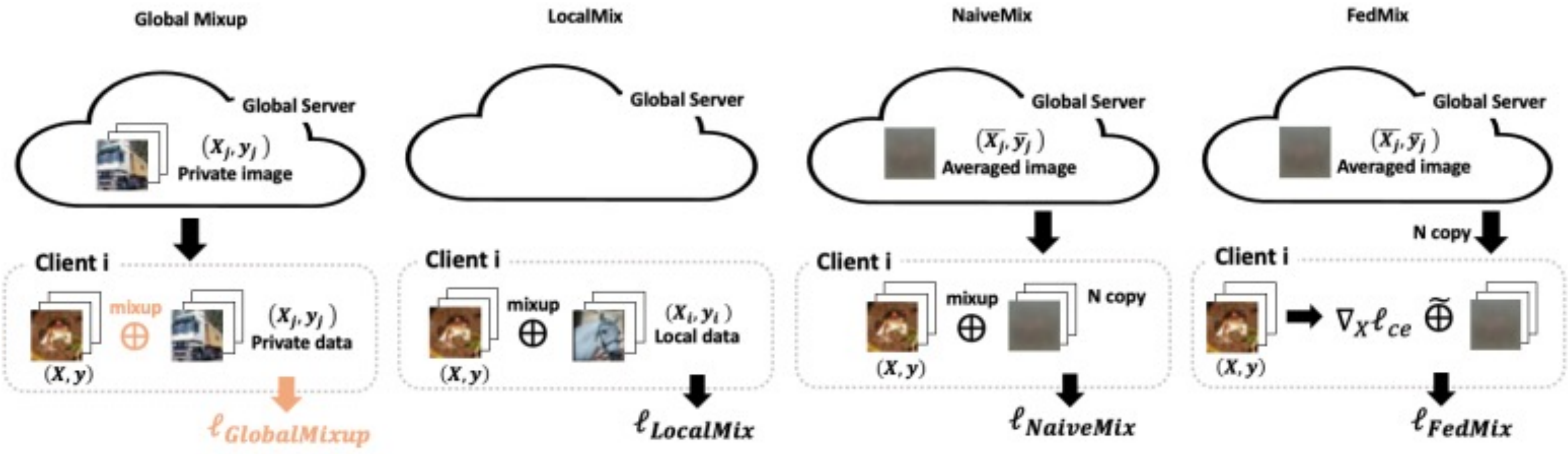
Information Sensitivity: General\External

Source: https://arxiv.org/pdf/1602.05629.pdf

Federated Mixup (FedMix)

Information Sensitivity: General\External

FedMix - introduction



(a) Global Mixup (b) Local Mixup (c) NaiveMix (d) FedMix

Figure 1: Brief comparisons of Mixup strategies in FL and MAFL. (a) Global Mixup: Raw data is exchanged and directly used for Mixup between local and received data, which violates privacy. (b) Local Mixup: Mixup is only applied within client's local data. (c) NaiveMix: Under MAFL, Mixup is performed between local data and received *averaged* data. (d) FedMix: Under MAFL, our novel algorithm approximates Global Mixup using input derivatives and *averaged* data.

Source: https://arxiv.org/pdf/1907.09693.pdf

Information Sensitivity: General\External

FedMix - algorithm

Algorithm 1: Mean Augmented Federated Learning (MAFL)

Input: $\mathbb{D}_k = \{\boldsymbol{X}_k, \boldsymbol{Y}_k\}$ for $k = 1, \ldots, N$
M_k: number of data instances used for computing average $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$

Initialize w_0 for global server
for $t = 0, \ldots, T-1$ **do**
  for *client* k *with updated local data* **do**
    Split local data into M_k sized batches
    Compute $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ for each batch
    Send all $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ to server
  end
  $\mathbb{S}_t \leftarrow K$clients selected at random
  Send $\boldsymbol{w}_t$ to clients $k \in \mathbb{S}_t$
  if *updated* **then**
    Aggregate all $\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}$ to $\boldsymbol{X}_g, \boldsymbol{Y}_g$
    Send $\boldsymbol{X}_g, \boldsymbol{Y}_g$ to clients $k \in \mathbb{S}_t$
  end
  for $k \in \mathbb{S}_t$ **do**
    $\boldsymbol{w}_{t+1}^k \leftarrow \mathit{LocalUpdate}(k, \boldsymbol{w}_t; \boldsymbol{X}_g, \boldsymbol{Y}_g)$
  end
  $\boldsymbol{w}_{t+1} \leftarrow \frac{1}{K} \sum_{k \in \mathbb{S}_t} p_k \boldsymbol{w}_{t+1}^k$
end

Algorithm 2: FedMix

$\mathit{LocalUpdate}(k, \boldsymbol{w}_t; \boldsymbol{X}_g, \boldsymbol{Y}_g)$ under MAFL (Algorithm 1):
$\boldsymbol{w} \leftarrow \boldsymbol{w}_t$
for $e = 0, \ldots, E-1$ **do**
  Split $\mathbb{D}_k$ into batches of size B
  for *batch*$(\boldsymbol{X}, \boldsymbol{Y})$ **do**
    Select an entry $\boldsymbol{x}_g, \boldsymbol{y}_g$ from $\boldsymbol{X}_g, \boldsymbol{Y}_g$
    $\ell_1 = (1-\lambda)\ell\big(f((1-\lambda)\boldsymbol{X}; \boldsymbol{w}), \boldsymbol{Y}\big)$
    $\ell_2 = \lambda\ell\big(f((1-\lambda)\boldsymbol{X}; \boldsymbol{w}), \boldsymbol{y}_g\big)$
    $\ell_3 = \lambda\frac{\partial \ell_1}{\partial \boldsymbol{x}} \cdot \boldsymbol{x}_g$
    (derivative calculated at $\boldsymbol{x} = (1-\lambda)\boldsymbol{x}_i$ and $y = y_i$ for each of $\boldsymbol{x}_i, y_i$ in $\boldsymbol{X}, \boldsymbol{Y}$)
    $\ell = \ell_1 + \ell_2 + \ell_3$
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta_{t+1}\nabla\ell$
  end
end
return $\boldsymbol{w}$

Source: https://arxiv.org/pdf/1907.09693.pdf

Information Sensitivity: General\External

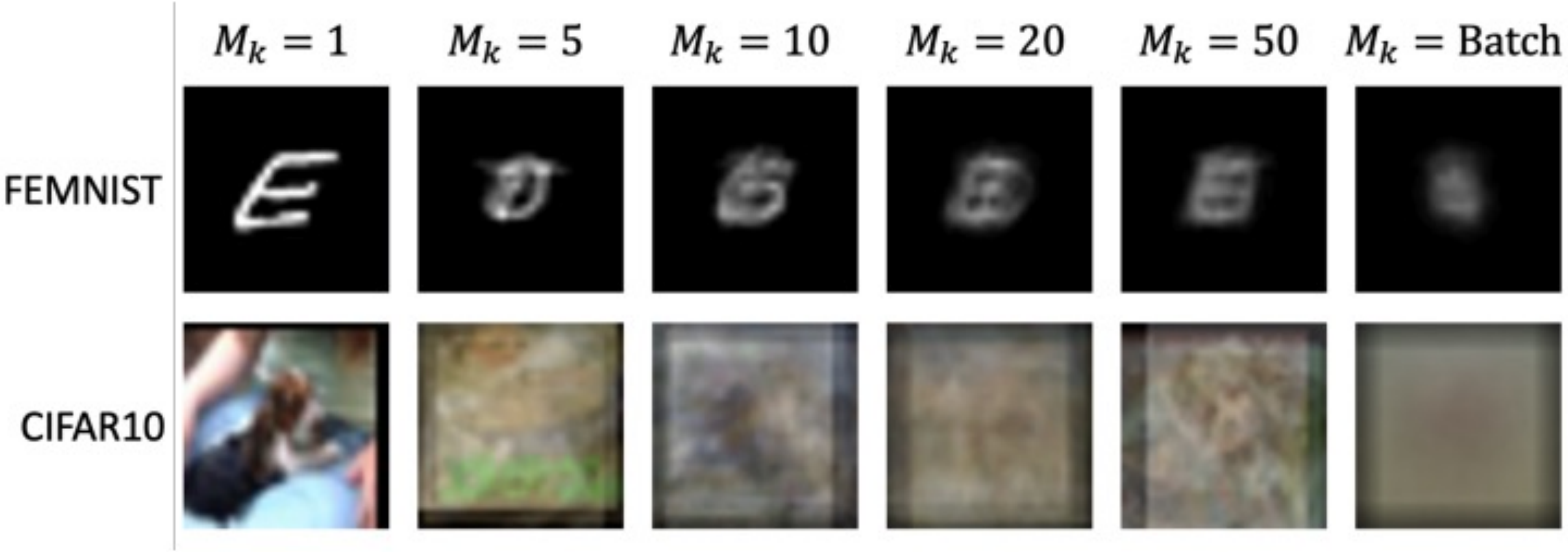
FedMix - results

Table 1: **Test accuracy after (target rounds)** and **number of rounds to reach (target test accuracy)** on various datasets. Algorithms in conjunction with FedProx are compared separately (bottom). MAFL-based algorithms are marked in bold.

Algorithm	FEMNIST		CIFAR10		CIFAR100	
	test acc. (200)	rounds (80%)	test acc. (500)	rounds (70%)	test acc. (500)	rounds (40%)
Global Mixup	88.2	8	88.2	85	61.4	54
FedAvg	85.3	26	73.8	283	50.4	101
LocalMix	82.8	28	73.0	267	54.8	91
NaiveMix	85.9	23	77.4	198	53.8	85
FedMix	**86.5**	**18**	**81.2**	**162**	**56.7**	**34**
FedProx	84.6	**29**	77.3	266	51.2	79
FedProx + LocalMix	84.1	39	74.1	314	54.0	90
FedProx + **NaiveMix**	85.7	37	76.7	230	53.1	74
FedProx + **FedMix**	**86.0**	32	**78.9**	**223**	**54.5**	**63**

Source: https://arxiv.org/pdf/1907.09693.pdf

FedMix – How did passed information look like?



Source: https://arxiv.org/pdf/1907.09693.pdf

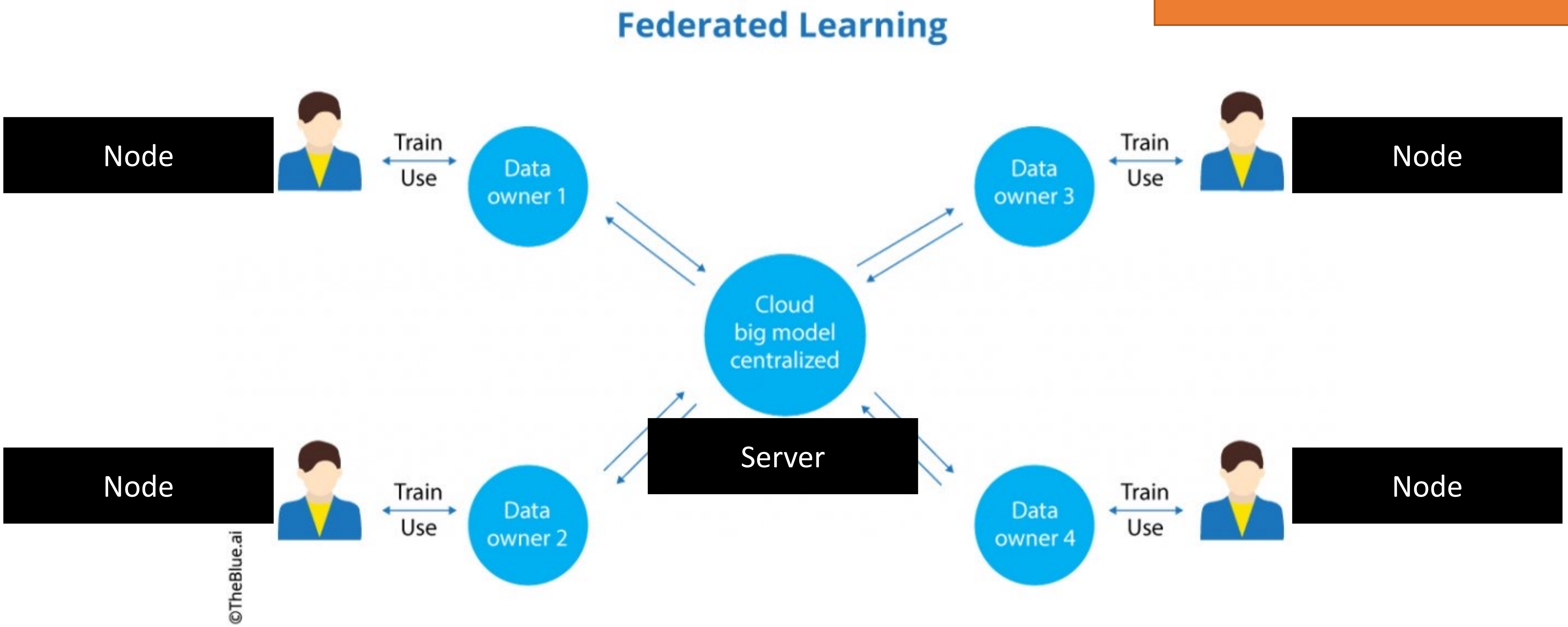
Information Sensitivity: General\External

StatMix – ICONIP 2022 – method presentation

Information Sensitivity: General\External

REPEATED

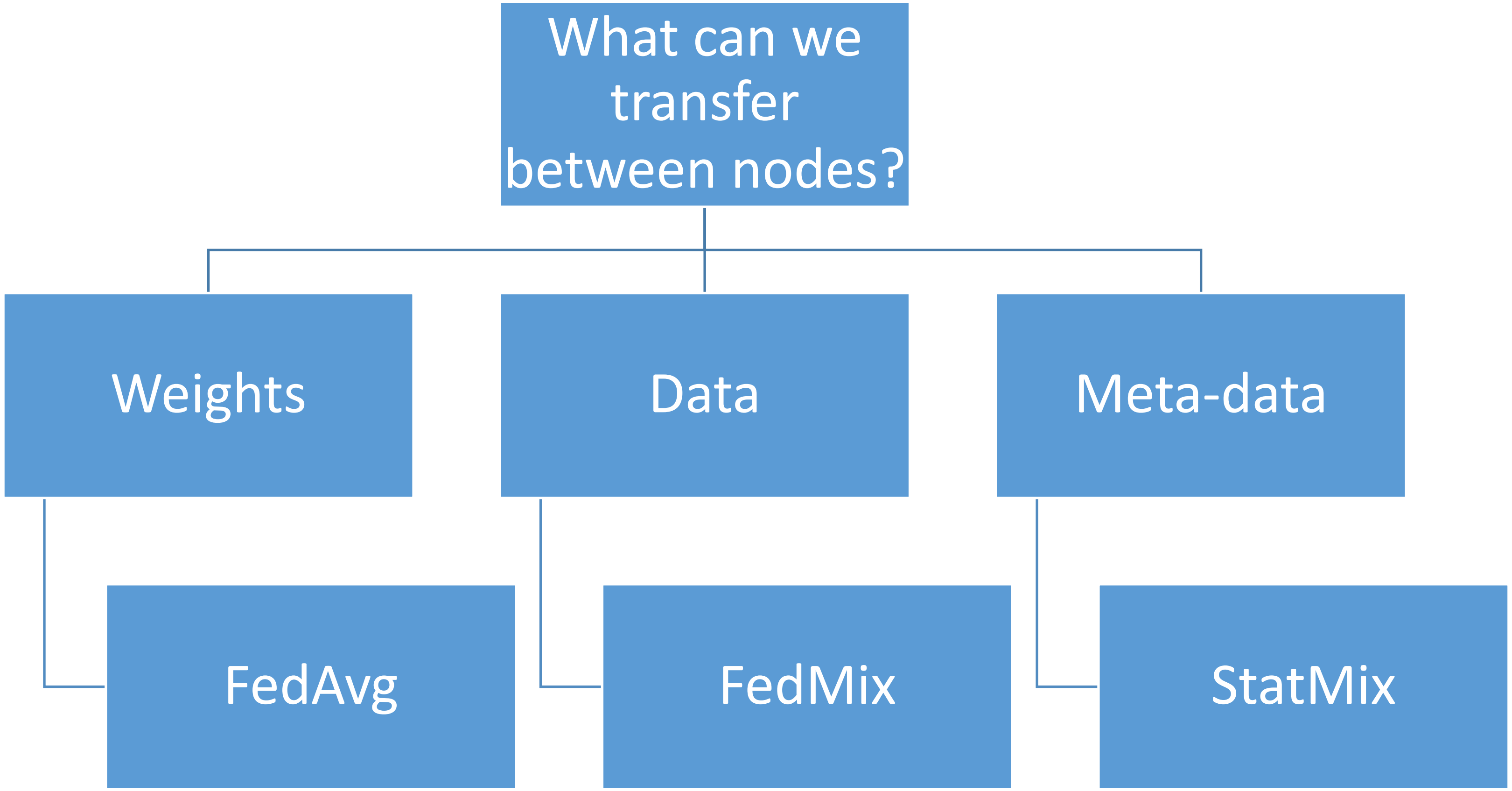
Federated Learning – introduction and general notation



Source: https://www.google.com/url?sa=i&url=https%3A%2F%2Ftheblue.ai%2Fblog-pl%2Ffederated-learning-2%2F&psig=AOvVaw2HJ0aM7PRIJV9lkdL-W9WX&ust=1665899766885000&source=images&cd=vfe&ved=0CA0QjRxqFwoTCMj5ibXG4foCFQAAAAAdAAAAABAI

REPEATED

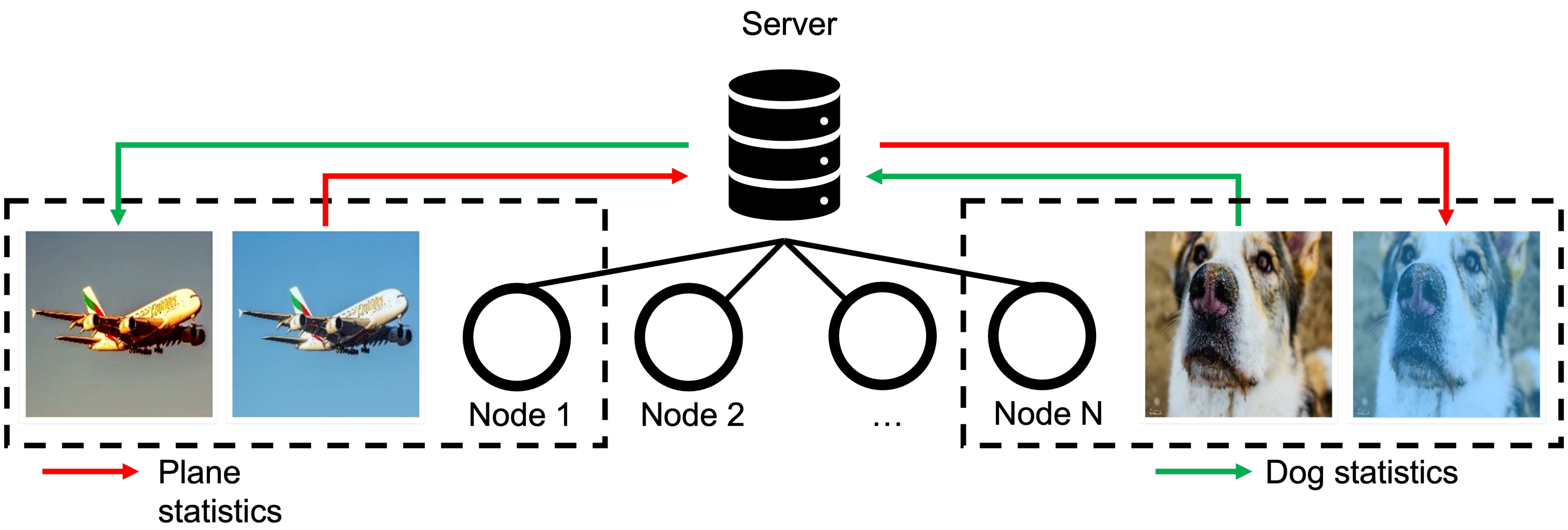
Approaches to augmentation in Federated Learning



Source: https://arxiv.org/pdf/1907.09693.pdf

Information Sensitivity: General\External

StatMix – How it works?



Source: https://arxiv.org/pdf/2207.04103.pdf

StatMix – How it works?

Preparation phase

Algorithm 1 *StatMix*

Local part 1:

$K \leftarrow$ number of images in the node; $N \leftarrow$ number of nodes
for $i = 1, 2, \ldots, N$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    Calculate all the image statistics according to equations (1)-(2)
    $S_{ik} = \{\mu(x_{ik})_1, \mu(x_{ik})_2, \mu(x_{ik})_3, \sigma(x_{ik})_1, \sigma(x_{ik})_2, \sigma(x_{ik})_3\}$
  end for
end for
Share statistics with the sever

Sever part:

Distribute statistics to all nodes

$$\mu(x_{ik})_c = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} x_{ik}[w, h, c] \quad (1)$$

$$\sigma(x_{ik})_c = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \left(x_{ik}[w, h, c] - \mu(x_{ik})_c\right)^2} \quad (2)$$

where $x_{ik}[w, h, c]$ is a value of $[w, h]$ pixel of image x_{ik}, in color channel c.

Information Sensitivity: General\External

Source: https://arxiv.org/pdf/2207.04103.pdf

StatMix – How it works?

Preparation phase

Algorithm 1 *StatMix*

Local part 1:

$K \leftarrow$ number of images in the node; $N \leftarrow$ number of nodes
for $i = 1, 2, \ldots, N$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    Calculate all the image statistics according to equations (1)-(2)
    $S_{ik} = \{\mu(x_{ik})_1, \mu(x_{ik})_2, \mu(x_{ik})_3, \sigma(x_{ik})_1, \sigma(x_{ik})_2, \sigma(x_{ik})_3\}$
  end for
end for
Share statistics with the sever

Sever part:

Distribute statistics to all nodes

Local part 2:

for $i = 1, 2, \ldots, N$ **do**
  for $epoch = 1, 2, \ldots, max\_epoch$ **do**
    for $batch = 1, 2, \ldots, max\_batch$ **do**
      if $random(0, 1) < P_{StatMix}$ **then**
        Randomly select set of statistics $S_{jm}, j \in \{1, \ldots, N\}, m \in \{1, \ldots, K\}$
        Normalize images from a batch using equation (3)
        Apply augmentation using equation (4)
      end if
    end for
  end for
end for

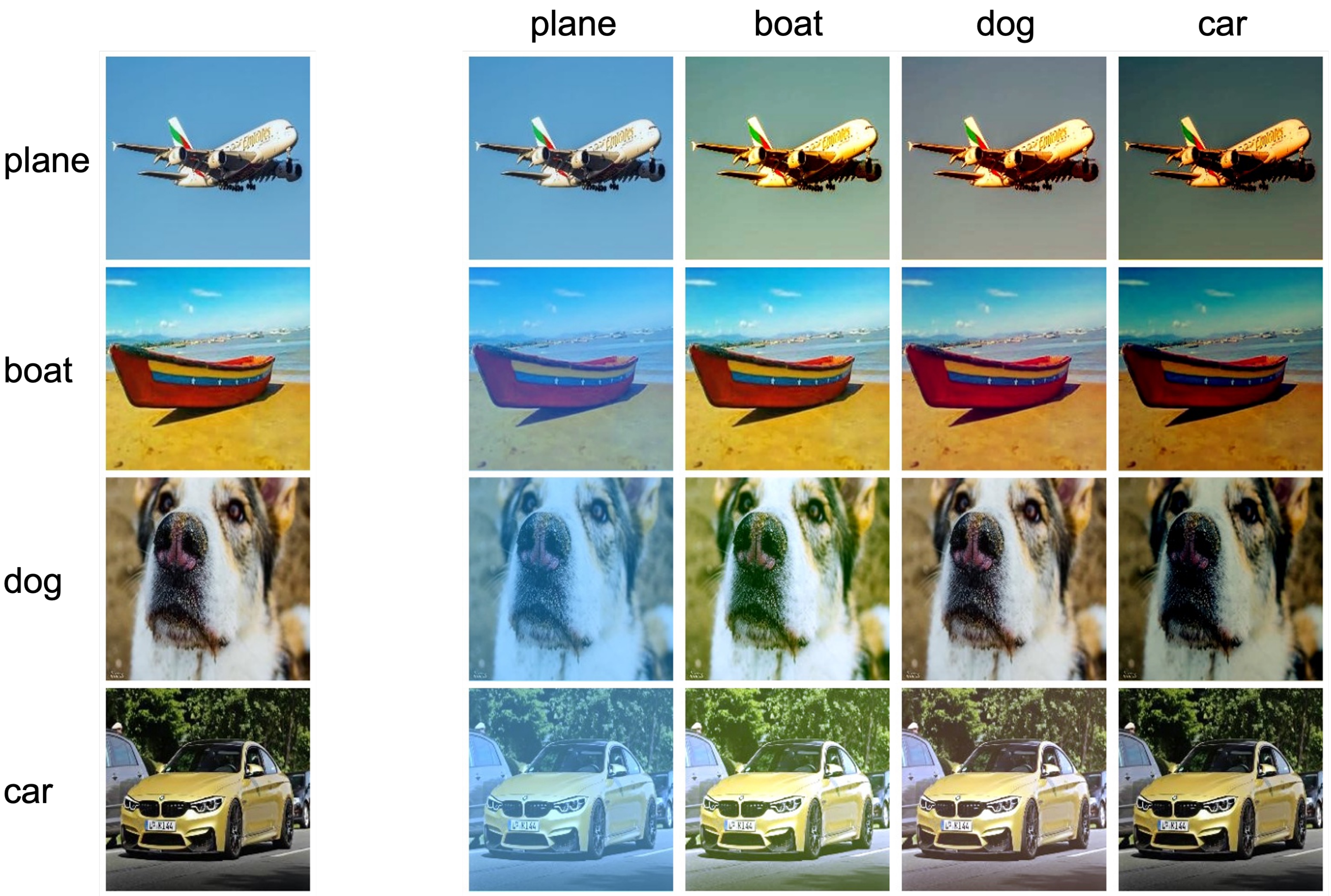
Execution phase

Source: https://arxiv.org/pdf/2207.04103.pdf

Information Sensitivity: General\External

StatMix – How it looks?

Input images: Resulting images, statistics from:



Source: https://arxiv.org/pdf/2207.04103.pdf

StatMix – How does it perform?

How to read the table?

Table 1. Mean and standard deviation results for CIFAR-10 dataset averaged over last 10 epochs and 3 experiment repetitions. Columns denote: number of nodes (N), model architecture, whether or not standard DA was applied, whether *StatMix* augmentation was used (0.0 – not used, 0.5 – used with probability 0.5), the relative improvement of applying *StatMix* compared to not applying it, i.e. [mean(0.5) / mean(0.0) – 1].

			StatMix				
			0.0		0.5		
Nodes (N)	Architecture	Standard	mean	std	mean	std	diff [%]
1	DLA	False	86.02	0.80	86.58	0.47	0.65
		True	93.26	0.28	93.83	0.19	0.61
	PreActResNet18	False	86.15	0.79	86.60	0.14	0.52
		True	93.54	0.05	93.79	0.13	0.27
5	DLA	False	67.32	1.15	69.47	0.70	3.19
		True	63.39	1.03	66.24	0.89	4.50
	PreActResNet18	False	70.83	0.44	72.01	0.55	1.67
		True	68.22	0.64	69.12	0.33	1.32
10	DLA	False	56.06	1.27	58.97	1.09	5.19
		True	50.72	1.45	54.54	1.59	7.53
	PreActResNet18	False	60.72	0.64	62.03	0.76	2.16
		True	56.63	0.77	58.69	0.74	3.64
50	DLA	False	37.47	1.20	38.06	1.42	1.57
		True	34.06	1.11	34.65	1.39	1.73
	PreActResNet18	False	38.62	0.96	40.28	1.08	4.30
		True	35.01	1.07	36.93	1.21	5.48

Source: https://arxiv.org/pdf/2207.04103.pdf

Information Sensitivity: General\External

StatMix – How does it perform?

Works also for non-federated scenario!

Table 1. Mean and standard deviation results for CIFAR-10 dataset averaged over last 10 epochs and 3 experiment repetitions. Columns denote: number of nodes (N), model architecture, whether or not standard DA was applied, whether *StatMix* augmentation was used (0.0 – not used, 0.5 – used with probability 0.5), the relative improvement of applying *StatMix* compared to not applying it, i.e. [mean(0.5) / mean(0.0) – 1].

			StatMix				
			0.0		0.5		
Nodes (N)	Architecture	Standard	mean	std	mean	std	diff [%]
1	DLA	False	86.02	0.80	86.58	0.47	0.65
		True	93.26	0.28	93.83	0.19	0.61
	PreActResNet18	False	86.15	0.79	86.60	0.14	0.52
		True	93.54	0.05	93.79	0.13	0.27
5	DLA	False	67.32	1.15	69.47	0.70	3.19
		True	63.39	1.03	66.24	0.89	4.50
	PreActResNet18	False	70.83	0.44	72.01	0.55	1.67
		True	68.22	0.64	69.12	0.33	1.32
10	DLA	False	56.06	1.27	58.97	1.09	5.19
		True	50.72	1.45	54.54	1.59	7.53
	PreActResNet18	False	60.72	0.64	62.03	0.76	2.16
		True	56.63	0.77	58.69	0.74	3.64
50	DLA	False	37.47	1.20	38.06	1.42	1.57
		True	34.06	1.11	34.65	1.39	1.73
	PreActResNet18	False	38.62	0.96	40.28	1.08	4.30
		True	35.01	1.07	36.93	1.21	5.48

Source: https://arxiv.org/pdf/2207.04103.pdf

Information Sensitivity: General\External

StatMix – How does it perform?

Impact depends on the complexity of the task and network.

Table 1. Mean and standard deviation results for CIFAR-10 dataset averaged over last 10 epochs and 3 experiment repetitions. Columns denote: number of nodes (N), model architecture, whether or not standard DA was applied, whether *StatMix* augmentation was used (0.0 – not used, 0.5 – used with probability 0.5), the relative improvement of applying *StatMix* compared to not applying it, i.e. [mean(0.5) / mean(0.0) – 1].

			StatMix				
			0.0		0.5		
Nodes (N)	Architecture	Standard	mean	std	mean	std	diff [%]
1	DLA	False	86.02	0.80	86.58	0.47	0.65
		True	93.26	0.28	93.83	0.19	0.61
	PreActResNet18	False	86.15	0.79	86.60	0.14	0.52
		True	93.54	0.05	93.79	0.13	0.27
5	DLA	False	67.32	1.15	69.47	0.70	3.19
		True	63.39	1.03	66.24	0.89	4.50
	PreActResNet18	False	70.83	0.44	72.01	0.55	1.67
		True	68.22	0.64	69.12	0.33	1.32
10	DLA	False	56.06	1.27	58.97	1.09	5.19
		True	50.72	1.45	54.54	1.59	7.53
	PreActResNet18	False	60.72	0.64	62.03	0.76	2.16
		True	56.63	0.77	58.69	0.74	3.64
50	DLA	False	37.47	1.20	38.06	1.42	1.57
		True	34.06	1.11	34.65	1.39	1.73
	PreActResNet18	False	38.62	0.96	40.28	1.08	4.30
		True	35.01	1.07	36.93	1.21	5.48

Source: https://arxiv.org/pdf/2207.04103.pdf

Information Sensitivity: General\External

StatMix – How does it perform?

Table 2. Mean and standard deviation results for CIFAR-100 dataset, average from 10 epochs and 3 experiment repetitions. Columns, denote: number of nodes (N), model architecture, whether or not standard DA was applied, whether *StatMix* augmentation was used (0.0 – not used, 0.5 – used with probability 0.5), relative improvement of applying *StatMix* compared to not applying it, i.e. [mean(0.5) / mean(0.0) – 1].

			StatMix				
			0.0		0.5		
Nodes (N)	Architecture	Standard	mean	std	mean	std	diff [%]
1	DLA	False	59.29	2.08	58.11	0.87	-1.99
		True	73.40	0.26	75.25	0.46	2.52
	PreActResNet18	False	54.99	2.73	55.84	2.21	1.55
		True	71.83	0.49	73.63	0.22	2.51
5	DLA	False	26.46	0.49	28.04	0.53	5.97
		True	22.84	0.71	24.84	0.60	8.76
	PreActResNet18	False	31.02	0.58	31.39	0.58	1.19
		True	27.70	0.60	28.63	0.59	3.36
10	DLA	False	19.86	0.59	20.49	0.66	3.17
		True	16.48	0.57	17.80	0.92	8.01
	PreActResNet18	False	22.32	0.41	22.86	0.50	2.42
		True	19.37	0.50	20.33	0.57	4.96
50	DLA	False	9.65	0.64	9.56	0.72	-0.93
		True	7.83	0.69	7.77	0.74	-0.77
	PreActResNet18	False	10.74	0.46	10.48	0.56	-2.42
		True	9.15	0.45	9.20	0.48	0.55

Source: https://arxiv.org/pdf/2207.04103.pdf

Information Sensitivity: General\External

StatMix – How does it perform?

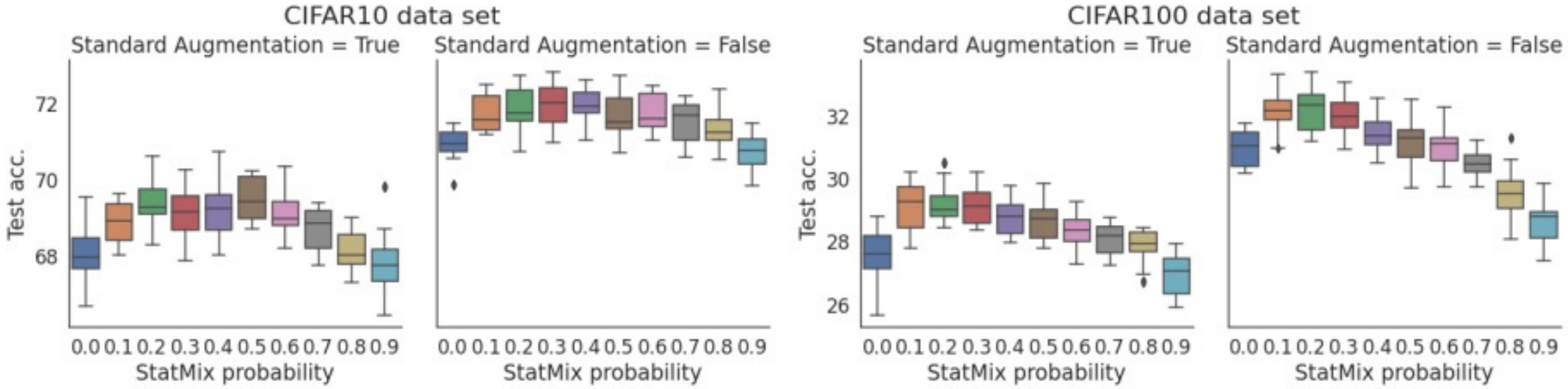


Fig. 3. CIFAR10 and CIFAR100 test accuracy as a function of probability of applying *StatMix* in FL setup with 5 nodes ($N = 5$) on PreActResNet18 architecture. The values are averaged over last 10 epochs and 3 independent experiment repetitions. For each dataset the left figure refers to experiments that utilize standard input DA, the right one presents results without its application.

Information Sensitivity: General\External

Source: https://arxiv.org/pdf/2207.04103.pdf

StatMix – Coclusion

- A simplistic data augmentation (DA) mechanism (**StatMix**), dedicated to FL learning setup that limits the amount of communication between participating nodes, is proposed.

- **StatMix** is evaluated on two different CNNs, with numbers of FL nodes ranging from 5 to 50, and shows promising results, improving baseline by between 0.3% and 7.5% depending on the architecture and the number of nodes.

- It is shown that the standard set of simple DAs, typically used for CIFAR datasets, is not well suited for FL scenario, as it deteriorates the performance along with a decrease of the number of samples per each FL node.

Source: https://arxiv.org/pdf/2207.04103.pdf

Information Sensitivity: General\External