ON THE BYZANTINE ROBUSTNESS OF CLUSTERED FEDERATED LEARNING

Felix Sattler[1]*, Klaus-Robert Müller*[2,3,4]*, Thomas Wiegand*[1]*, Wojciech Samek*[1]

[1] Fraunhofer Heinrich Hertz Institute, Berlin, Germany
[2] TU Berlin, Berlin, Germany, [3] Max Planck Institute for Informatics, Saarbrücken, Germany
[4] Korea University, Seoul, South Korea

ABSTRACT

Federated Learning (FL) is currently the most widely adopted framework for collaborative training of (deep) machine learning models under privacy constraints. Albeit it's popularity, it has been observed that Federated Learning yields suboptimal results if the local clients' data distributions diverge. The recently proposed Clustered Federated Learning Framework addresses this issue, by separating the client population into different groups based on the pairwise cosine similarities between their parameter updates. In this work we investigate the application of CFL to byzantine settings, where a subset of clients behaves unpredictably or tries to disturb the joint training effort in an directed or undirected way. We perform experiments with deep neural networks on common Federated Learning datasets which demonstrate that CFL (without modifications) is able to reliably detect byzantine clients and remove them from training.

Index Terms— robust learning, distributed learning, Federated Learning, multi-task learning, clustering

1. INTRODUCTION

Federated Learning is a distributed training framework, which allows multiple clients (typically mobile or IoT devices) to jointly train a single deep learning model on their combined data in a communication-efficient way, without requiring any of the participants to reveal their private training data to a centralized entity or to each other [1][2][3][4][5]. Federated Learning relies on the assumption that one single model $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ is able to fit all client's data generating distributions φ_i at the same time. This assumption can be formally stated as follows:

Assumption 1. ***("Federated Learning"):*** *There exists a parameter configuration $\theta^* \in \Theta$, that (locally) minimizes the risk on all clients' data generating distributions at the same time*

$$R_i(\theta^*) \leq R_i(\theta) \; \forall \theta \in B_\varepsilon(\theta^*), i = 1, .., m \tag{1}$$

Hereby $R_i(\theta) = \int l(f_\theta(x), y) d\varphi_i(x, y)$ is the risk function associated with distribution φ_i and l is a suitable loss function.

It is easy to see that this assumption is not always satisfied. Concretely it is violated if either (a) clients have disagreeing conditional distributions $\varphi_i(y|x) \neq \varphi_j(y|x)$ or (b) the model f_θ is not expressive enough to fit all distributions at the same time. Simple counter examples for both cases are presented in Figure 3 in the Appendix. In the following we will call two clients and their distributions φ_i and φ_j *congruent* (with respect to f and l) if they satisfy Assumption 1 and *incongruent* if they don't.

The recently proposed Clustered Federated Learning Framework (CFL) [6] generalizes the Federated Learning Assumption 1 and is able to solve Federated Learning problems where the clients hold data from incongruent distributions which adhere to a clustering structure:

Assumption 2. ***("Clustered Federated Learning"):*** *There exists a partitioning $\mathcal{C} = \{c_1, .., c_k\}$, $\bigcup_{i=1}^{k} c_k = \{1, .., m\}$ of the client population, such that every subset of clients $c \in \mathcal{C}$ satisfies the conventional Federated Learning Assumption.*

Clustered Federated Learning (Fig. 1) exploits geometric properties of the FL loss surface, to identify the clustering structure $\mathcal{C}$. In contrast to existing Federated Multi-Task Learning approaches, CFL does not require any modifications to the FL communication protocol to be made, is applicable to general non-convex objectives (in particular deep neural networks) and comes with strong mathematical guarantees on the clustering quality. We will quickly recap the theoretical foundations of CFL in Section 2.

The Byzantine Setting: In has been shown that regular Federated Learning fails to converge in the presence of faulty and malicious clients [9][10]. One single bad client can compromise the performance of the entire jointly trained model, and negate the training efforts of all other clients. To mitigate this

This work was supported by the German Ministry for Education and Research as Berlin Big Data Center (01IS14013A) and the Berlin Center for Machine Learning (01IS18037I). Partial funding by DFG is acknowledged (EXC 2046/1, project-ID: 390685689). This work was also supported by the Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-00451).

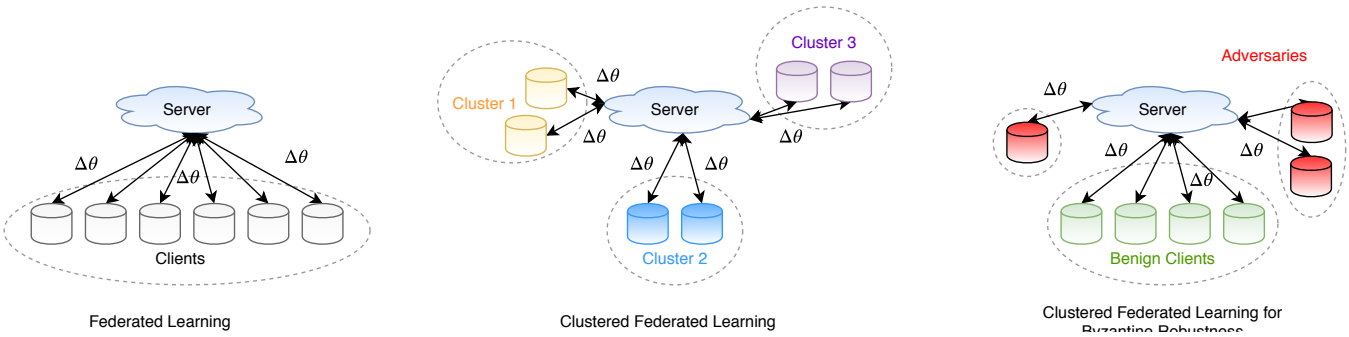

Fig. 1: Clustered Federated Learning (center) is an extension to the conventional Federated Learning framework (left), which increases robustness and flexibility by automatically separating clients into clusters of jointly trainable data distribution. In this work we investigate the application of CFL to the byzantine setting (right), which can be viewed as a special case of the clustered setting, where only the largest cluster is considered benign and all other clusters are considered adversarial.

problem, different robust federated learning strategies have been proposed in the literature. However, these existing strategies require modifications to the Federated communication protocol to be made and are often computationally expensive.

In this paper we will explore the application of CFL to these byzantine settings, where a subset of the client population behaves in an explicitly harmful manner. It is easy to see, that these settings can be subsumed under Assumption 2 by declaring one particular cluster of clients $c_{benign} \in \mathcal{C}$ as the "benign" clients and all other clients, which are incongruent to this cluster as "adversarial": $\mathcal{C} = \{c_{benign}\} \cup \mathcal{C}_{adv}$.

2. CLUSTERED FEDERATED LEARNING

For general Federated Learning problems which follow Assumption 2 it has been shown in [6] that Clustered Federated Learning is able to infer the clustering structure $\mathcal{C}$ under relatively mild conditions on the clients and their data by solely inspecting the cosine similarity

$$\alpha_{i,j} := \frac{\langle \Delta\theta_i, \Delta\theta_j \rangle}{\|\Delta\theta_i\| \|\Delta\theta_j\|} \tag{2}$$

between the different clients' gradient updates. The result can be summarized in the following theorem:

Theorem 1 (Separation Theorem + Corollary [6])**.** *Let $D_1, .., D_m$ be the local training data of m different clients, each dataset sampled from one of k different data generating distributions $\varphi_1, .., \varphi_k$, such that $D_i \sim \varphi_{I(i)}(x, y)$. Let the empirical risk $r_i(\theta)$ on every client approximate the true risk $R_{I(i)}(\theta)$ at every stationary solution of the Federated Learning objective θ^* s.t.*

$$\gamma_i := \frac{\|\nabla R_{I(i)}(\theta^*) - \nabla r_i(\theta^*)\|}{\|\nabla R_{I(i)}(\theta^*)\|} \in [0, 1) \tag{3}$$

Then, at every stationary solution θ^* of the Federated Learning objective, the bi-partitioning

$$c_1, c_2 \leftarrow \arg\min_{c_1 \cup c_2 = c} \left(\max_{i \in c_1, j \in c_2} \alpha_{i,j} \right). \tag{4}$$

will always be correct if it holds that

$$\max_{i \in c_1, j \in c_2} \left[\cos\left(\frac{\pi}{k-1}\right) H_{i,j} + \sin\left(\frac{\pi}{k-1}\right) \sqrt{1 - H_{i,j}^2} \right] < \min_{\substack{i \neq j \\ I(i) = I(j)}} H_{i,j} \tag{5}$$

with

$$H_{i,j} = -\gamma_i \gamma_j + \sqrt{1 - \gamma_i^2} \sqrt{1 - \gamma_j^2}. \tag{6}$$

In this paper we obtain a slightly weaker, yet significantly simpler statement based on the maximum approximation error γ_{max} by simplifying condition (5):

Corollary 1. *As long as it holds that*

$$\gamma_{max} := \max_{i=1,..,m} \gamma_i < \sin\left(\frac{\pi}{4(k-1)}\right) \tag{7}$$

the clustering mechanism in equation (4) *will always produce a correct clustering.*

The proof of Corollary 1 is presented in the appendix.

Remark 1. *The fulfillment of inequality* (7) *is sufficient, but not necessary for obtaining a correct clustering. In practice, a correct clustering can be achieved with probability ≈ 1 for a much larger range of values of γ_{max} and k as detailed in [6].*

Algorithm: The Clustered Federated Learning Algorithm separates the client population in a top-down way: Starting from an initial set of clients $c = \{1, .., m\}$ and a parameter initialization θ_0, the clients compute weight-updates $\Delta\theta_i$ using n iterations of stochastic gradient descent. Like in conventional

Algorithm 1: Clustered Federated Learning (for Byzantine Robustness)

input: initial parameters θ_0, dissimilarity threshold $\alpha_{cross}^{max} \in [-1, 1)$, number of local iterations/ epochs n
outout: benign clients c_{benign}, benign model $\theta_{c_{benign}}$
init: set initial clusters $\mathcal{C} = \{\{1, .., m\}\}$, set initial models $\theta_i \leftarrow \theta_0 \; \forall i = 1, .., m$, set initial update $\Delta\theta_c \leftarrow 0 \; \forall c \in \mathcal{C}$
while *not converged* **do**
  for $i = 1, .., m$ ***in parallel*** **do**
    Client i does:
    - $\theta_i \leftarrow \theta_i + \Delta\theta_{c(i)}$
    - $\Delta\theta_i \leftarrow \text{SGD}_n(\theta_i, D_i) - \theta_i$
  end
  Server does:
  for $c \in \mathcal{C}$ **do**
    - $\alpha_{i,j} \leftarrow \frac{\langle \Delta\theta_i, \Delta\theta_j \rangle}{\|\Delta\theta_i\| \|\Delta\theta_j\|} \; \forall i, j \in c$
    - $c_1, c_2 \leftarrow \arg\min_{c_1 \cup c_2 = c} (\max_{i \in c_1, j \in c_2} \alpha_{i,j})$
    - $\alpha_{cross} \leftarrow \max_{i \in c_1, j \in c_2} \alpha_{i,j}$
    if $\alpha_{cross} < \alpha_{cross}^{max}$ **then**
      regular CFL:
      - $\mathcal{C} \leftarrow \mathcal{C} \setminus c \cup c_1 \cup c_2$

      byzantine robust CFL:
      - $\mathcal{C} \leftarrow \{\arg\max_{c \in \{c_1, c_2\}} |c|\}$ (10)
    end
  end
  for $c \in \mathcal{C}$ **do**
    - $\Delta\theta_c \leftarrow \frac{1}{|c|} \sum_{i \in c} \Delta\theta_i$
  end
end
return $\theta_c, \forall c \in \mathcal{C}$

Federated Learning, these weight-updates are then communicated to a centralized server where they are aggregated into a global model update. In CFL however, prior to model aggregation the server computes the matrix of pairwise cosine-similarities α and, based on this, the cluster candidates

$$c_1, c_2 \leftarrow \arg \min_{c_1 \cup c_2 = c} (\max_{i \in c_1, j \in c_2} \alpha_{i,j}). \tag{8}$$

If the similarity between the two clusters

$$\alpha_{cross} \leftarrow \max_{i \in c_1, j \in c_2} \alpha_{i,j} \tag{9}$$

is is below a certain threshold α_{cross}^{thresh}, then the main cluster is bi-partitioned into the two cluster candidates. Model aggregation is then performed separately for every cluster of clients. In the subsequent communication rounds all existing clusters can potentially be split up further using the same mechanism. The entire procedure is presented in Algorithm 1.

The Byzantine Setting: In principle, the above described Algorithm for CFL does not need to be modified for the byzantine setting. However, as we assume in the byzantine setting that the majority of clients belongs to one single benign cluster and all other clients are considered adversarial, we can of course save computation effort by excluding all clients from training, which do not belong to the largest cluster, via (10).

3. RELATED WORK

Conventional Federated Learning [1][2] has been extensively investigated in congruent non-iid scenarios [7][8], but is unable to deal with the challenges of incongruent data distributions in general and adversarial clients in particular as argued in section 1. Adversarial Federated Learning settings can be roughly organized into two groups: In byzantine settings it is assumed that a subset of the client population behaves in an arbitrary (random) manner. This setting has been extensively studied and a variety of robust aggregation rules have been proposed which rely on gradient similarity [9][10], geometric median aggregation [11], redundant communication [12] or adaptive model quality estimation [13]. While some of these proposed methods offer convergence guarantees in the byzantine setting, they are also expensive in terms of computation or communication and often require modifications to the Federated communication protocol. A more difficult problem setting is the one of Federated Learning poisoning. In this setting, (possibly multiple) clients try to introduce a hidden back-door functionality into the jointly trained model [14][15][16][17]. As the adversaries in this setting are allowed to adapt their attacks based on the model updates they receive from the server, they are much harder to detect and to this day no efficient defense strategies have been proposed.

4. EXPERIMENTS

We adopt the experimental setup from [13] to investigate the robustness of Clustered Federated Learning against byzantine and adversarial clients. We perform experiments on the well-known MNIST, Fashion-MNIST and CIFAR-10 data sets on which we train convolutional deep neural networks using SGD with a batch-size of 100. A detailed specification of data sets and models can be found in the Appendix. Like [13] we consider two different Federated Learning settings, one with 10 and one with 100 participating clients among which we split the training data randomly and evenly. In each experiment we declare 30% of the client population to be faulty/ malicious. We consider three different scenarios: 1.) In the **Byzantine** scenario, the malicious clients draw their weight-updates $\Delta\theta$ from a centered Gaussian distribution with isotropic covariance matrix and standard deviation 1 (instead of computing them using stochastic gradient descent). This scenario simulates either an undirected attack against the collaborative training procedure. 2.) In the **Label-Flip** scenario, all the labels of the training data for the malicious clients are set to zero. This scenario simulates a directed attack, with the goal

Table 1: Accuracy achieved by conventional Federated Learning and CFL in the four investigated scenarios.

		Byzantine	Noisy	Label-Flip	Clean
MNIST	FL	9.8%	96.9%	91.3%	97.5
	CFL (ours)	**93.19%**	**97.4%**	**97.4%**	97.4%
Fashion-MNIST	FL	9.6%	77.12%	60.6%	79.9
	CFL (ours)	**78.0%**	**79.7%**	**79.7%**	80.2
CIFAR	FL	0.1	70.4%	40.1	76.0
	CFL (ours)	/	**74.6%**	**74.7%**	75.3%

to disproportionally bias the jointly trained model towards one specific class. 3.) In the **Noisy** scenario, the faulty clients train on data $\hat{x} = \mathcal{U}(-10, 10)$ that is made of uniform noise. This scenario simulates unstructured noisy distortions of the training data. 4.) In the **Clean** scenario, no adversaries are present. A good robust training algorithm should not harm the convergence in this scenario.

In each scenario, we perform Clustered Federated Learning according to Algorithm 1 over the entire client population and set the threshold for the necessary cosine dissimilarity between different clusters to $\alpha_{cross}^{thresh} = 0.02$.

Our goal is to investigate whether CFL as defined in Algorithm 1 is able to detect the faulty and malicious clients in the above scenarios and remove them from the main cluster. Figure 2 shows the development of α_{cross} over the first 200 communication rounds for 70 benign and 30 malicious clients on our three data sets and three different scenarios. Every time the value of α^{cross} falls below $\alpha_{cross}^{thresh} = 0.02$ the clients are separated into two different groups according to equation (4). These events are marked in the plot as follows: Whenever adversarial clients are separated from the main cluster of benign clients this is marked green in the plot and the number of adversarial clients that were separated in this particular round is plotted. Whenever benign clients are removed from the main cluster this is marked red in the the plot.

As we can see, on all data sets and different adversarial scenarios it takes less than 40 communication rounds for all malicious clients to be removed from training. In the noisy and label-flip scenario the adversarial clients all share the same data generating distribution which causes them to be separated out all at once by CFL. In the byzantine scenario where the adversarial clients communicate random normal updates every adversary forms it's own cluster and hence it takes several clustering rounds to filter out all adversaries. In the clean scenario the cross-cluster similarity α_{cross} never falls below the threshold and hence the client population is never separated. Clustered Federated Learning hence does not harm the convergence in the clean scenarion.

In every performed experiment all of the malicious clients are separated out first and fully. The final accuracy achieved by FL and CFL after 200 rounds of communication is shown in Table 1. As we can see CFL always achieves higher accuracy than regular FL, with the accuracy difference being the largest in the byzantine and label-flip scenarios.

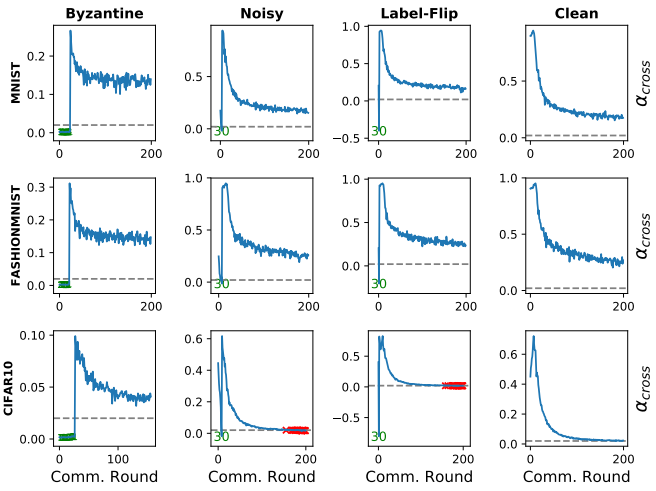

Fig. 2: Development of the cluster candidate dissimilarity α^{cross} over the first 200 communication rounds of Clustered Federated Learning. Whenever the value of α^{cross} dips below $\alpha_{max}^{cross} = 0.02$ the main cluster is separated into two according to equation 4. A correct clustering is displayed in greed and incorrect clustering is displayed in red. If the clustering is correct the number of correctly separated clients is shown in the plot. On all three data sets and adversarial scenarios all 30 adversaries are separated out within at most 34 communication rounds.

5. CONCLUSION

Clustered Federated Learning is a recently proposed extension to the regular Federated Learning framework, which increases robustness and flexibility by automatically separating clients into clusters of jointly trainable data distribution. In this work, we investigated the application of CFL to byzantine scenarios which can be viewed as a special case of divergent client data distribution where one particular distribution is declared to be the benign distribution and all other distributions are considered adversarial. In experiments on three different data sets and with three different types of adversarial scenarios we find that CFL (without any modifications) is capable of filtering out adversarial clients within relatively few communication rounds. This demonstrates that CFL can offer significant advantages over regular Federated Learning even in situations where clients do not form an obvious clustering structure.

6. REFERENCES

[1] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al., "Communication-efficient learn-

ing of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[2] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[3] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth, "Practical secure aggregation for privacy preserving machine learning.," *IACR Cryptology ePrint Archive*, vol. 2017, pp. 281, 2017.

[4] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al., "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[5] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith, "Federated learning: Challenges, methods, and future directions," *arXiv preprint arXiv:1908.07873*, 2019.

[6] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek, "Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints," *arXiv preprint arXiv:1910.01991*, 2019.

[7] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek, "Robust and communication-efficient federated learning from non-iid data," *arXiv preprint arXiv:1903.02891*, 2019.

[8] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[9] Peva Blanchard, Rachid Guerraoui, Julien Stainer, et al., "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, pp. 119–129.

[10] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault, "The hidden vulnerability of distributed learning in byzantium," *arXiv preprint arXiv:1802.07927*, 2018.

[11] Yudong Chen, Lili Su, and Jiaming Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 44, 2017.

[12] Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," *arXiv preprint arXiv:1803.09877*, 2018.

[13] Luis Muñoz-González, Kenneth T Co, and Emil C Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," *arXiv preprint arXiv:1909.05125*, 2019.

[14] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 27–38.

[15] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[16] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov, "How to backdoor federated learning," *arXiv preprint arXiv:1807.00459*, 2018.

[17] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo, "Analyzing federated learning through an adversarial lens," *arXiv preprint arXiv:1811.12470*, 2018.

7. SUPPLEMENT

7.1. Proof of Corollary 1

Proof. By Theorem 1 we know that equation (4) will always produce a correct clustering if

$$0 < \alpha_{intra}^{min} - \alpha_{cross}^{max} \tag{11}$$

By monotonicity of $H_{i,j}$ we have

$$H_{i,j} = -\gamma_i\gamma_j + \sqrt{1-\gamma_i^2}\sqrt{1-\gamma_j^2} \tag{12}$$

$$\geq 1 - 2\gamma_{max}^2 \tag{13}$$

Therefore

$$\alpha_{intra}^{min} = \min_{\substack{i\neq j \\ I(i)=I(j)}} H_{i,j} \geq 1 - 2\gamma_{max}^2 \tag{14}$$

and

$$\alpha_{cross}^{max} = \max_{i\in c_1, j\in c_2} [\cos(\frac{\pi}{k-1})H_{i,j} + \sin(\frac{\pi}{k-1})\sqrt{1-H_{i,j}^2}] \tag{15}$$

$$\leq \cos(\frac{\pi}{k-1})(1-2\gamma_{max}^2) + \sin(\frac{\pi}{k-1})\sqrt{1-(1-2\gamma_{max}^2)^2} \tag{16}$$

After abbreviating

$$c := \cos(\frac{\pi}{k-1}),\ s := \sin(\frac{\pi}{k-1}) \tag{17}$$

we therefore know that equation (4) will produce a correct clustering if

$$1 - 2\gamma_{max}^2 - c(1-2\gamma_{max}^2) - s\sqrt{1-(1-2\gamma_{max}^2)^2} > 0 \tag{18}$$

which is equivalent to

$$2s\gamma_{max}\sqrt{1-\gamma_{max}^2} < (1-c)(1-2\gamma_{max}^2) \tag{19}$$

As both sides of the equation are greater than zero this is equivalent to

$$4s^2\gamma_{max}^2(1-\gamma_{max}^2) < (1-c)^2(1-2\gamma_{max}^2)^2 \tag{20}$$

After substituting $x := \gamma_{max}^2$ and re-arranging terms we get

$$0 < (1-c)^2(1-4x+4x^2) - 4s^2x + 4s^2x^2 \tag{21}$$

$$= x^2(4(1-c)^2 + 4s^2) + x(-4(1-c)^2 - 4s^2) + (1-c)^2 \tag{22}$$

$$= x^2(4(1-2c+c^2+s^2)) - x(4(1-2c+c^2+s^2)) \tag{23}$$

$$+ (1-c^2) \tag{24}$$

By the trigonometric equality it holds that $s^2 + c^2 = 1$ and the term can be further simplified to

$$0 < 8(1-c)x^2 - 8(1-c)x + (1-c)^2 \tag{25}$$

which is equivalent to

$$0 < x^2 - x + \frac{1-c}{8} \tag{26}$$

The latter inequality is satisfied if and only if

$$x \gtrless \frac{1}{2} \pm \sqrt{\frac{1+c}{8}} \tag{27}$$

which can be further simplified using the trigonometric equality

$$1 + \cos(z) = 2\cos^2(\frac{z}{2}) \tag{28}$$

and we get

$$x \gtrless \frac{1}{2} \pm \sqrt{\frac{\cos^2(\frac{\pi}{2(k-1)})}{4}} \tag{29}$$

This is equivalent to

$$x \gtrless \frac{1 \pm \cos(\frac{\pi}{2(k-1)})}{2} \tag{30}$$

and hence

$$\gamma_{max} \gtrless \sqrt{\frac{1 \pm \cos(\frac{\pi}{2(k-1)})}{2}} \tag{31}$$

Since we required

$$\gamma_{max} < \sin(\frac{\pi}{4(k-1)}) \leq \sqrt{\frac{1}{2}} \tag{32}$$

only the solution

$$\gamma_{max} < \sqrt{\frac{1-\cos(\frac{\pi}{2(k-1)})}{2}} \tag{33}$$

is valid. Using the trigonometric equality

$$1 - \cos(z) = 2\sin^2(\frac{z}{2}) \tag{34}$$

we finally end up with

$$\gamma_{max} < \sin(\frac{\pi}{4(k-1)}). \tag{35}$$

□

Table 2: Accuracy achieved by conventional Federated Learning and CFL on the three benchmarks.

		Byzantine	Noisy	Label-Flip	None
MNIST	FL	9.8%	98.1%	81.6%	98.6
	CFL (ours)	95.5	98.4%	98.6%	98.5
Fashion-MNIST	FL	9.6%	83.0%	68.4%	85.1
	CFL (ours)	86.1%	85.4%	84.9	85.1
CIFAR	FL	/	73.7%	/	77.65
	CFL (ours)	/	74.3%	/	76.0%

7.2. Experiments

Datasets and Models:

The MNIST dataset contains 60000 28×28 grey-scale training images and 10000 test images of handwritten digits in 10 categories. We train a four-layer convolutional neural net with relu activations. The architecture is as follows:

```
conv [1,32,5,5] -> ReLU -> maxpool [2,2]
-> conv [32,32,5,5] -> ReLU -> maxpool [2,2]
-> fc [800,256] -> ReLU -> fc [256,10]
```

The model is trained using conventional SGD with a learning-rate of 0.01 and no momentum.

The Fashion-MNIST dataset contains 60000 grey-scale training images and 10000 test images of fashion items in 10 categories. We train the same four-layer convolutional neural net with relu activations as for MNIST.

The CIFAR-10 data set contains 50000 rgb training images and 10000 test images of natural and human made objects in 10 categories. We train a simplified version of the popular VGG network, with 8 convolutional layers followed by 3 fully connected layers and relu activations. The architecture is as follows (with every convolutional and fully connected layer followed by a ReLU activation):

```
conv [3,32,3,3] -> maxpool [2,2]
-> conv [32,64,3,3] -> maxpool [2,2]
-> conv [64,128,3,3] -> conv [128,128,3,3]
-> maxpool [2,2] -> conv [128,128,3,3]
-> conv [128,128,3,3] -> maxpool [2,2]
-> conv [128,128,3,3] -> conv [128,128,3,3]
-> maxpool [2,2] -> fc [128,128]
-> fc [128,128] -> fc [128, 10]
```

The model is trained using momentum SGD with a learning-rate of 0.01 and a momentum coefficient of 0.9.

7.3. Additional Results

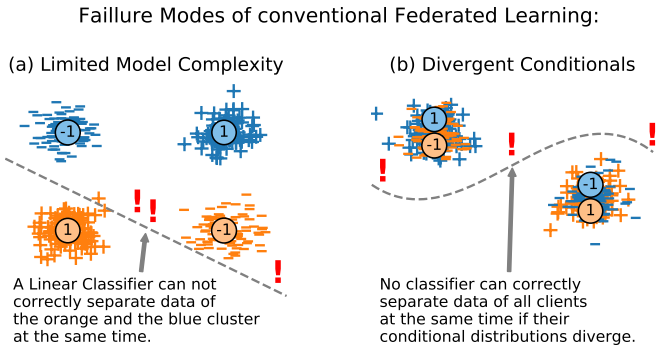

Fig. 3: Two toy cases in which the Federated Learning Assumption is violated. Blue points belong to clients from the first cluster while orange points belong to clients from the second cluster. Left: Federated XOR-problem. An insufficiently complex model is not capable of fitting all clients' data distributions at the same time. Right: If different clients' conditional distributions diverge, *no model* can fit all distributions at the same time. In both cases the data on clients belonging to the same cluster can be easily separated.

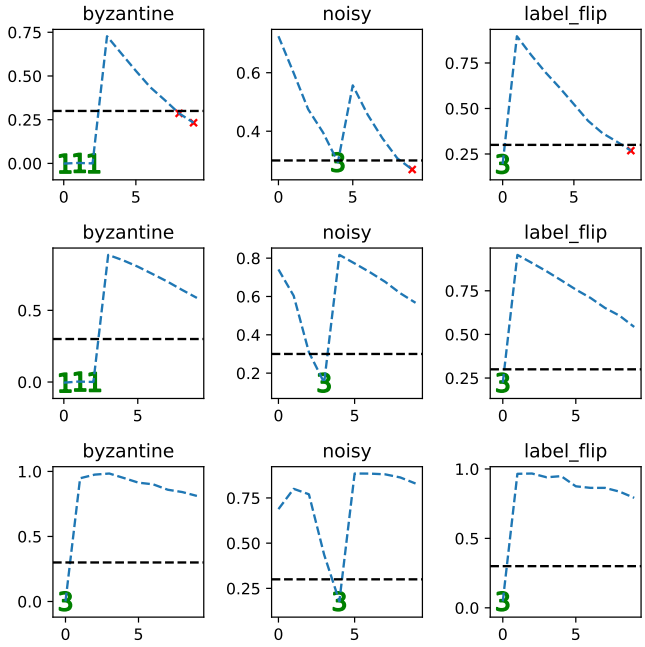

Fig. 4