# Certified Federated Adversarial Training

**Giulio Zizzo**
IBM Research Europe
giulio.zizzo2@ibm.com

**Ambrish Rawat**
IBM Research Europe
ambrish.rawat@ie.ibm.com

**Mathieu Sinn**
IBM Research Europe
mathsinn@ie.ibm.com

**Sergio Maffeis**
Imperial College London
sergio.maffeis@imperial.ac.uk

**Chris Hankin**
Imperial College London
c.hankin@imperial.ac.uk

## Abstract

In federated learning (FL), robust aggregation schemes have been developed to protect against malicious clients. Many robust aggregation schemes rely on certain numbers of benign clients being present in a quorum of workers. This can be hard to guarantee when clients can join at will, or join based on factors such as idle system status, and connected to power and WiFi [12]. We tackle the scenario of securing FL systems conducting adversarial training when a quorum of workers could be completely malicious. We model an attacker who poisons the model to insert a weakness into the adversarial training such that the model displays *apparent* adversarial robustness, while the attacker can exploit the inserted weakness to bypass the adversarial training and force the model to misclassify adversarial examples. We use abstract interpretation techniques to detect such stealthy attacks and block the corrupted model updates. We show that this defence can preserve adversarial robustness even against an adaptive attacker.

## 1 Introduction

Training via federated learning (FL) [14] is increasingly popular due to the many strengths of FL, which include reducing communication overheads, decentralising computations, and preserving data privacy. However, FL offers new attack opportunities to an adversary. Attackers can participate in FL and pursue objectives such as inserting backdoors [23] or preventing model convergence [2]. To combat this, robust aggregation schemes have been developed [3, 26, 25, 24]. These algorithms offer protection if the proportion of malicious clients does not exceed particular thresholds. For example, median based robust aggregation schemes require the majority of the clients participating in a round of FL to be benign [26]. However, having this assurance in practical settings at all times can be difficult to maintain: benchmark FL systems [4] have many thousands of users, of which only a small proportion participate in any one FL round. Therefore, if clients are modelled as randomly participating, or worse if they can join at will, then it is likely that a round of FL will occur in which the quorum of participating clients is maliciously dominated breaking robust aggregation defences.

In parallel, it is known that neural networks are vulnerable to adversarial examples, which in turn can be mitigated using adversarial training [13]. The interaction of adversarial training with FL is an active area of research with results showing federated adversarial training's sensitivity to the amount of local compute [17], that not all clients need to necessarily perform adversarial training to achieve robustness [11], as well as specialised attacks against federated adversarial training [27].

In this work we consider how federated adversarial training could be conducted if malicious clients can dominate particular quorums. Specifically, the scenario we consider is:

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.

- A defender conducts FL in a large distributed setting, and only a small number of clients participate in any one FL round. The defender cannot guarantee that a particular quorum of workers is not maliciously dominated.
- The defender seeks a robust model and so instructs the clients to conduct adversarial training.

We consider two research questions arising from this scenario: how can a defender protect themselves against stealthy compromise of their model? Additionally, to what extent can an adaptive attacker overcome a proposed defense? The contributions we make in addressing the above questions are:

- Demonstrate the effectiveness of certification based on abstract interpretation techniques as a defensive method in federated learning. Due to the guarantees that abstract interpretation can provide, stealthy model tampering can be instantly detected.
- A novel attack strategy that can be employed by an adaptive attacker to try and evade certification defences. The attack exploits the potentially limited number of datapoints a defender can verify. So, the tampered model that the attacker sends only certifies the datapoints that the defender checks while still being stealthily weak on all other data.

## 2 Background

### 2.1 Federated Learning

In FL many clients wish to learn a common model, and each client trains using their local data. Once the clients have updated their local models they send their trained weights to an aggregator. The aggregator combines the received weights producing an updated model. The model is then sent to the clients for a new training round. However, malicious clients can propose arbitrary updates pursuing objectives such as preventing convergence [2] or inserting backdoors [23]. To that end robust aggregation schemes have been developed to filter out malicious updates [3, 26, 25, 24].

Adversarial training for federated learning is vulnerable to convergence attacks [27]. However, we focus on a more dangerous kind of attack [27], where an adversary replaces a model which is undergoing adversarial training with one that is only superficially robust due to gradient masking. In this case, the defender can be unaware that the model has been compromised.

Robust aggregation defences rely on bounding the number of malicious clients $k$ present in a quorum of $n$ clients. Median based defences handle at at most $\lceil \frac{n}{2} \rceil - 1$ malicious clients [26], Krum needs $n > 2k + 2$ [3], or Bulyan [10] requires $n \geq 4k + 3$. However, benchmark datasets for simulating real FL have a very small proportion of clients participating in a round [4]. This raises difficulties for such defences as, even if there is a small proportion of adversaries in a system, it is likely that given sufficient training rounds, there will be rounds in which the threshold requirements for robust aggregation schemes are broken. Therefore, to offer protection in this challenging environment, we turn to certification based approaches.

### 2.2 Certification

Certification of neural networks is a research area orthogonal to FL. Via certifiable methods a defender has a *guarantee* whether a datapoint can have its prediction changed which can be used to uncover stealthy model compromise. More precisely, given a neural network $f$ the defender wishes to verify that the predicted label for all inputs from a $p$-norm ball $B^p_\epsilon$ of size $\epsilon$ centred on the datapoint $x$ are classified to the same class $T$,

$$\text{arg max} f(x') = T \qquad \forall x' \in B^p_\epsilon(x) = \{x' = x + \delta \mid \ |\delta|_p \leq |\epsilon|_p\}. \tag{1}$$

If such a condition holds then a datapoint is certified. We also refer to certifying neural networks by measuring the proportion of datapoints it can certifiably classify. There are several research strands which try and verify adversarial robustness [8, 20] through certification. In this work we model a defender using abstract interpretation methods to verify the adversarial robustness of the model. With abstract interpretation we pass through the neural network a datapoint's abstract representation in a domain such as interval [9], zonotope [18], octagon [15], or polyhedra [19]. More precisely, we can represent a concrete input $x$ with all its possible perturbations with an abstract element $\hat{x}$. The abstract element will capture the entirety of $B^p_\epsilon$, and by passing $\hat{x}$ through the neural network, and analysing its abstract output $\hat{y}$, we can check if a robustness property holds for the network.

The various abstract representations often trade precision for scalability. The zonotope domain was shown to offer strong performance while remaining computationally tractable for neural networks of realistic size, and is the domain we will focus on [18]. A zonotope $\hat{x}$ is represented by

$$\hat{x} = \eta_0 + \sum_{i=1}^{i=N} \eta_i \epsilon_i \tag{2}$$

where $\eta_0$ corresponds to a central coefficient, $\eta_i$ representing $i = 0 \dots N$ deviations around the center with associated error symbols $\epsilon_i$. The values $\epsilon_i$ can take either +1 or -1. Pushing zonotopes through neural networks requires computing *abstract transformers* for various arithmetic operations. Many operations, like matrix multiplication, can be carried to the abstract domain easily. However, certain operations require more consideration. For the neural networks we consider, ReLUs can cause the analysis to become more imprecise. To balance scalability and precision in our abstract transformer for ReLUs we use the DeepZono formulation proposed in [18]. To help build intuition for the reader into using zonotopes for neural networks we show a toy example in Appendix A.

**Certification in FL:** There have begun to be several works which examine certifying a FL trained model against backdoors. Parallel work to ours [22] examined parameter smoothing to certify robustness against $L_2$ backdoors. General provable robustness was investigated in [5], which although protects against *any* type of adversary, it required training hundreds of parallel models with tight requirements on the number of malicious clients. In contrast, centralised training was examined in [21] for certification against backdoors via training several models on noise corrupted data. This poses problems in FL as malicious clients may not conduct the prescribed training procedure.

**Certified Training:** It is worth considering the case in FL where the clients perform certified training [16, 1] rather than normal adversarial training. We assume the defender does not pursue this for a few reasons. Firstly, performing certified training causes a drop in accuracy on normal data: this is on top of the drop in accuracy that occurs with projected gradient descent (PGD) training [1]. Additionally, certified training methods can result in worse white-box PGD performance [16]. Recent work [1] is bridging the gap, but conducting certified training in FL is an open challenge. Secondly, certified training itself has many open questions which should be addressed prior to making it federated such as the conflict between more accurate abstractions leading to worse certifiable results [1].

Having established the relevant background we now specify our attacker model more precisely, detail our proposed defence, and finally discuss adaptive attacks against it.

### 2.3 Attacker Model

**Attacker Objective:** We consider an attacker who wishes to stealthily compromise the adversarial training procedure. Specifically, they wish their compromised model to have the following properties:

- High *normal* data performance.
- High adversarial robustness when the model is evaluated with standard application of PGD to the maximum allowable $L_\infty$ budget that the defender will evaluate to.
- A hidden weakness that the attacker can exploit such that they can drive the adversarial performance of the model down when still limited to modifying the data within the given $L_\infty$ budget.

There are several ways the above could be achieved. Firstly, a backdoor with a $L_\infty$ bounded trigger could be inserted. The attacker will train the model adversarially, but also insert the backdoor. When the model is presented with the backdoor trigger, even if it was adversarially trained, the model will misclassify to an attacker chosen class. Backdoor examples, with discussion on $L_\infty$ backdoors, are shown in Appendix B. Alternatively, a stronger style of attack is to pursue gradient masking based strategies such as sending a defensively distilled model to the aggregator [27]. The defender, unaware that they have a model with defensive distillation will see, through standard application of PGD, that the model is apparently robust. An attacker, who deliberately sent the defensively distilled model, can then bypass the brittle defence by application of temperature scaling to the softmax [6]. As the distilled model does not have a backdoor trigger it can sidestep many methods which rely on backdoor detection and certification [22]. A review of defensive distillation and its use as an attack method is outlined in Appendix C.

**Attacker Knowledge and Capabilities:** The malicious clients cannot determine when they participate in a FL round, but when participating they can co-ordinate between each other. For our basic adversary we assume the attacker has access to all the data held by the malicious clients, knows the robust aggregation rules, and knows the benign model updates that are sent to the aggregator.

## 3 Certified Defence

The defender wishes to conduct federated adversarial training and, being aware of potential attackers, does not blindly accept model updates and will employ a robust aggregation algorithm. In addition, the defender has access to a held out set of data (similar defender modelling assumptions were used in [25]) and rejects models which perform poorly on it, both in terms of normal and adversarial accuracy against standard white box PGD attack [13]. The new component we propose is a certification based assertion to detect stealthy compromise. More precisely, the defender checks:

1. The certified accuracy over a certification dataset which is drawn from a held out dataset. We define the certified accuracy as the proportion of data that satisfies the certification condition given by Equation 1, and in addition, is also classified correctly by the neural network.
2. Secondly, as we will later see, the certifiable accuracy is often not sensitive enough to distinguish between a corrupted vs adversarially trained model. As a more sensitive metric the defender will also use the certifiable loss. We use a modified version of the loss in [16] of,

   $$L(z, y) = \max(0, \max_{q \neq y}(z_q - z_y)) \tag{3}$$

   where $z_q$ is the component of the zonotope associated with the logit for label $q$ and $y$ is the target label. Computing the max operation over all the $z_q - z_y$ pairs can be computationally unfeasible as each zonotope component can have thousands of $\epsilon$ error terms. Hence, we approximate and compute the max operation over the interval representation of $z_q - z_y$.

Each round the defender will collect the relevant metrics (normal accuracy, adversarial accuracy, and mean certifiable loss and certified accuracy) and will accept the updates if they have at least 90% of the various accuracy scores from the prior round, and the certifiable loss has varied by a maximum of 10%. This is with the assumptions that attackers can only join randomly and so not coordinate over multiple rounds to slowly drive down model performance. More sophisticated filtering strategies can clearly build on this baseline.

If the aggregated model during a FL round passes all of these criteria then it will be accepted by the aggregator, else the round is skipped.

## 4 Adaptive Attacks Against Certification

We propose a novel adaptive attack which can be employed by a strong adversary who has knowledge of the certification defence to try and evade it. The certification algorithm itself is unassailable: by definition of the soundness of the abstract interpretation methods, it is impossible to have the certification algorithm claim a datapoint cannot be misclassified and then, through any attack algorithm, misclassify that datapoint. Hence, the attacker will need to consider alternative attack vectors.

The attack takes advantage of the fact that certification approaches can be slow and computationally expensive. The DeepZono transformation which we use is a *fast* certification, and yet can take a few seconds for complex data: in the original paper which proposed it [18] only the first 100 samples of the MNIST and CIFAR10 test datasets were used for the results. Hence, the attacker can exploit the limited number of samples that could be realistically checked by a defender. Still maintaining the original three attack objectives, we add an additional goal of:

- Match the certification metrics over the datapoints that the defender will check.

The adaptive attacker knows the exact certification set, in addition to any other data that the defender uses to check the model performance. This represents the strongest possible adversary with perfect knowledge. By evaluating our defence under this threat model we can obtain measures of its worst case performance. The attacker pursues their goal via the following methodology:

1. Train a model with defensive distillation. Such a model will fulfil the original goals defined in Section 2.3, but will be detected by the certifiable check as being vulnerable.
2. Refine the defensively distilled model obtained from Step 1. The attacker optimises for the model to match the certification statistics that a defender uses to detect stealthy model compromise: either certified accuracy or, in addition, the mean certifiable loss. However, just optimising for this objective removes the effect of defensive distillation. Therefore, the attacker jointly optimises to match the certifiable performance and maintain defensive distillation. The loss function to achieve both certifiable accuracy and match the mean certifiable loss is given by Equation 3.

Based on our experiments when developing this attack, this strategy does not work if the adversary pursued a backdoor based attack to bypass the adversarial training: the backdoor would be destroyed and a robust model would remain. We further experimented with instead conducting PGD training over the certification set in Step 2 to try and achieve the same certification metrics. However, this was unsuccessful. Although the model would classify the certification set correctly the certified accuracy and loss remained poor, and so the model would be detected by a defender as being compromised.

## 5 Experiments

We use the MNIST and CIFAR10 datasets. For both datasets we use 500 clients, each of which is randomly assigned a dataset partition. We set 175 clients to be malicious (35% of the total number). The neural networks used for each dataset are shown in Appendix D. In each round of training 5 clients are randomly selected to participate. The aggregator uses median based aggregation tolerating up to 2 malicious clients. We will be dealing with $L_\infty$ bounds in two different contexts: either the bound used for certification or the bound used to craft adversarial examples. To aid in disambiguation in the notation we use $L_\infty^{crt}$ for the bound to which certification is conducted, and $L_\infty^{adv}$ is the bound to which adversarial examples are constructed. For MNIST the models train adversarially with $L_\infty^{adv} = 0.25$, and for CIFAR10 we use $L_\infty^{adv} = 8/255$. The defender uses the first 1000 datapoints of the MNIST training dataset and the first 100 datapoints of the CIFAR10 training dataset as the certification datasets to certify the aggregated models. This is to have similar compute times spent on the certification defence with both datasets. Then, to compute the normal and adversarial accuracy scores in addition to the certification statistics each round (as described in Section 4) the defender has a validation dataset. For this, the defender uses the next 5000 points from the MNIST/CIFAR10 training sets. Neither the validation nor certification data is used for client training.

The attacker waits until the model has neared convergence and at least 3 malicious clients are selected to participate in a FL round. At that point the adversary has control of the system and runs the attack. The attacker runs either the distillation or backdoor attacks described in Section 2.3 to stealthily weaken the adversarial training. To match the PGD performance we use a temperature scaling for the defensive distillation of $T = 100$ for MNIST and $T = 25$ for CIFAR10. The backdoor trigger is composed of vertical stripes of magnitude $L_\infty = 0.1$ for MNIST and $6/255$ for CIFAR10 keeping the triggers below the adversarial training limit.

### 5.1 Certification in FL

We first demonstrate that against an non-adaptive attacker, a defender which cannot assure that the benign clients thresholds for robust aggregation schemes are met can use certification approaches to detect stealthy model compromise. The results for MNIST are shown in Table 1. Our first take-away is that relying solely on the certifiable accuracy as a metric gives a crude from of defence: although we adversarially trained with PGD at $L_\infty^{adv} = 0.25$ we do not have any certifiable accuracy at that level, and so leaves us blind to the differences between an honestly trained PGD model and the two compromised models. However, we *can* use the mean certifiable loss (Equation 3) over our certification points as a metric to distinguish between the models at $L_\infty^{adv} = 0.25$. At lower certification bounds (eg $L_\infty^{crt} = 0.1$ and $0.15$) a adversarially PGD trained model can have good certifiable accuracy. Hence, at those lower certification bounds both mean certifiable loss and accuracy differ significantly for an adversarially PGD trained model and the two attacker manipulated models.

| Dataset | Model | Acc | | Certified | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Normal | Adv | Acc | Loss | Acc | Loss | Acc | Loss |
| | | | $L_{\infty}^{adv} = 0.25$ | $L_{\infty}^{crt} = 0.1$ | | $L_{\infty}^{crt} = 0.15$ | | $L_{\infty}^{crt} = 0.25$ | |
| | PGD | 98.78 | 92.03 | 91.4 | 0.15 | 33.4 | 2.72 | 0.0 | 32.96 |
| MNIST | Distilled | 97.23 | 95.22 | 0.0 | 8.5x$10^3$ | 0.0 | 2.0x$10^4$ | 0.0 | 4.4x$10^4$ |
| | Backdoor | 98.81 | 94.10 | 0.0 | 50.81 | 0.0 | 96.47 | 0.0 | 202.43 |
| | | | $L_{\infty}^{adv} = 8/255$ | $L_{\infty}^{crt} = 1/255$ | | $L_{\infty}^{crt} = 4/255$ | | $L_{\infty}^{crt} = 8/255$ | |
| | PGD | 72.71 | 40.24 | 60.0 | 0.13 | 0.0 | 596.0 | 0.0 | 5.4x$10^3$ |
| CIFAR | Distilled | 78.35 | 44.17 | 0.0 | 2.1x$10^6$ | 0.0 | 9.5x$10^7$ | 0.0 | 3.7x$10^8$ |
| | Backdoor | 74.70 | 32.47 | 41.0 | 1.14 | 0.0 | 2.3x$10^3$ | 0.0 | 1.5x$10^4$ |

Table 1: At lower certification bounds large differences between a model trained via PGD compared to the distilled and backdoored models are visible in terms of both the certified accuracy and the mean loss across the certification set. At higher bounds only the mean certifiable loss is a viable metric.

For CIFAR10 we have a similar pattern: the certifiable accuracy is a good metric to distinguish between models at low certification bounds of $L_{\infty}^{crt} = 1/255$. When using the mean certifiable loss there is a large difference at all bounds making it simple to identify the tampered models.

### 5.2 Adaptive Attacker

We now examine an adaptive attacker who creates a brittle robustness model which certifies the defender's certification dataset. We use the strongest possible attacker model defined in Section 4 to push the defence as far as we can to examine its worst case performance. As discussed in Section 4, the backdoor attack is ineffective so an adversary uses the distillation attack and jointly optimises to match the certification statistics on the certification data. We show adaptive attack results for MNIST. For CIFAR10 the adaptive attack was unsuccessful: the classification task is more challenging, and optimising with the complex adversarial objective did not give a functioning model. Larger networks could have the capacity for such an objective. Such investigations are left for future work.

#### 5.2.1 Simpler Defender: Only Certifiable Accuracy Considered

We first consider a defender who just checks the certified accuracy of a model to demonstrate that this is only effective against an adaptive attacker at low $L_{\infty}^{adv}$ bounds. In Figure 1a the defender checks the certifiable accuracy at $L_{\infty}^{crt} = 0.1$, as at higher bounds such as $L_{\infty}^{crt} = 0.25$ the certifiable accuracy is not a distinguishing metric between a genuinely trained PGD model or a tampered model. We show the resulting performance when the adaptive attacker certifiably classifies varying amounts of the defender's certification dataset to $L_{\infty}^{crt} = 0.1$. The resulting model is evaluated with PGD white box attack with budgets ranging between $L_{\infty}^{adv} = 0.05 - 0.25$.

Each graph shows 1) the accuracy on normal clean data, 2) the accuracy on adversarial data with a temperature $T = 1$ on the softmax (no temperature effects) that the defender sees when evaluating with PGD, and 3) the performance of the model when attacked by an adversary using temperature scaling with $T = 100$ to bypass the defensive distillation. The adversarial examples crafted when using $T = 100$ are then evaluated on a model with $T = 1$ that a target client would be using.

We can see how the performance of the adaptive attack varies with the $L_{\infty}^{adv}$ budget for crafting adversarial examples. If the $L_{\infty}^{adv}$ budget is large in relation to the certified bound then the defence is ineffective. We see in Figure 1a the defender will see a model that fulfils their criteria of 1) high normal accuracy (green points), 2) high white box PGD performance (blue points) and 3) has the correct certification metrics. By applying temperature scaling when making an adversarial example, at high crafting budgets (eg dark red points), the model has poor actual adversarial robustness.

However, if the defender accepts lower levels of adversarial robustness then the adaptive attack boosts the underlying robustness of the model. As more samples are forced to be certifiably correct, then the adversarial robustness improves and generalises across the test set. Therefore, by considering certifiable accuracy, robustness to $L_{\infty}^{adv} = 0.25$ cannot be maintained against a adaptive attacker. However, this method does ensure robustness at lower bounds (e.g. $L_{\infty}^{adv} = 0.1$) *can* be achieved.

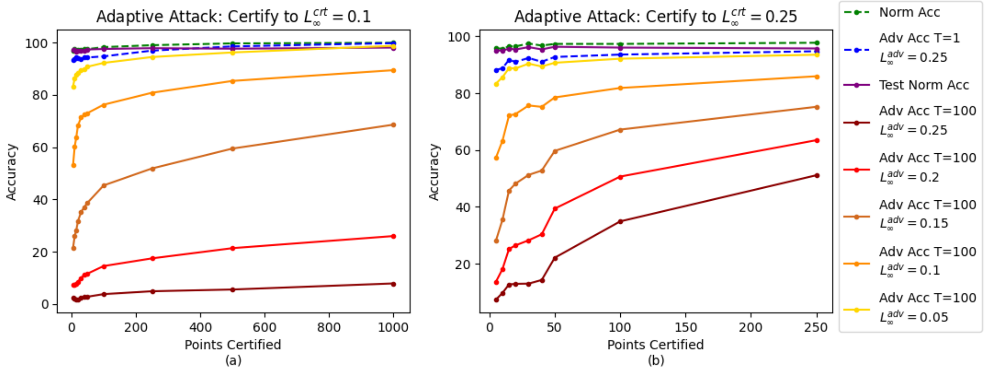


Figure 1: Performance of the adaptive attack. The green line shows the malicious model's performance on the clean validation set and the blue line represents what the defender sees when evaluating it with PGD on their validation set. The purple line is the accuracy of the malicious model on the clean MNIST test set. The yellow through to dark red hues represent the actual robustness of the model with differing $L_\infty^{adv}$ budgets on the MNIST test set. On the $x$ axis we have the number of datapoints from the defender's certification set that the attacker tries to either 1) certifiably classify in the left plot or 2) match the mean certifiable loss over in the right plot.

### 5.2.2 Full Defender: Certifiable Accuracy and Loss Considered

We now consider the more sensitive approach of using the certifiable loss which gives a signal at higher certification bounds to detect model compromise. We examine the performance when conducting certification on $L_\infty^{crt} = 0.25$ in Figure 1b. As a PGD trained model is unable to certifiably classify any of the certification set, the defender considers the mean certifiable loss which the attacker attempts to match. With a bound of $L_\infty^{crt} = 0.25$ conducting the adaptive attack is more challenging and immediately pursuing this with $L_\infty^{crt} = 0.25$ was ineffective. Instead we first certify the datapoints to a bound of $0.1$ and grow the certification bound in steps of $0.01$ when the attack succeeds in matching the mean certifiable loss.

The results are shown in Figure 1b. The adaptive attack was unable to proceed to larger certification sets: at 500 data points the attack stalled and did not find a solution after 72 hours of optimising. Further, the more datapoints the defender uses results in the neural network becoming *genuinely* robust if the attacker optimisation succeeds. By certifying an increasingly larger dataset the attacker is forced to impart real adversarial robustness into the model. Therefore, empirically the defence was robust with 1000 certification points for MNIST when subject to the strongest attacker we developed. Further, even if the attacker had unlimited computing speed and could certify larger datasets, from the trends in Figure 1b, the model's robustness will continue rising making the attacker less effective.

In addition to stalling at 500 certification points, the attack failed against CIFAR10. We postulate that due to the complexity of the attack balancing two contrasting objectives, the model capacity can be insufficient to solve the objectives. Therefore, the defender can "out compute" an adaptive attacker by certifying a number of datapoints for which an attacker is unable to optimise for. Future work would be to determine, for a given model, what the minimum certification set size is, such that even if the attacker optimisation succeeds, it results in a genuinely robust model. This is because the defender wishes to minimise the defence compute cost while ensuring that their model is safe.

## 6 Conclusion

We have examined the problem in FL of maliciously controlled quorums. There are many security problems that arise from this, and we consider the case of securing adversarial training. We have shown that certified methods form an effective defence. Adaptive attackers can force a manipulated model to have certified performance on a certification dataset and have hidden adversarial example weakness. However, this attack has a high compute cost making it impractical for more then a limited number of certification points. Additionally, with increasing certification data the attack strengthens the model's underlying adversarial robustness.

## References

[1] Mislav Balunovic and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations*, 2019.

[2] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8632–8642, 2019.

[3] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 119–129, 2017.

[4] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[5] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6885–6893, 2021.

[6] Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[7] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[8] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[9] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin T. Vechev. AI2: safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 3–18. IEEE Computer Society, 2018.

[10] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018.

[11] Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. Federated robustness propagation: Sharing adversarial robustness in federated learning. *arXiv preprint arXiv:2106.10196*, 2021.

[12] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[15] Antoine Miné. The octagon abstract domain. *Higher-order and symbolic computation*, 19(1):31–100, 2006.

[16] Matthew Mirman, Timon Gehr, and Martin T. Vechev. Differentiable abstract interpretation for provably robust neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3575–3583. PMLR, 2018.

[17] Devansh Shah, Parijat Dube, Supriyo Chakraborty, and Ashish Verma. Adversarial training in communication constrained federated learning. *arXiv preprint arXiv:2103.01319*, 2021.

[18] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin T Vechev. Fast and effective robustness certification. *NeurIPS*, 1(4):6, 2018.

[19] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019.

[20] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

[21] Maurice Weber, Xiaojun Xu, Bojan Karlaš, Ce Zhang, and Bo Li. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904*, 2020.

[22] Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. Crfl: Certifiably robust federated learning against backdoor attacks. *arXiv preprint arXiv:2106.08283*, 2021.

[23] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[24] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018.

[25] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, pages 6893–6901. PMLR, 2019.

[26] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018.

[27] Giulio Zizzo, Ambrish Rawat, Mathieu Sinn, and Beat Buesser. Fat: Federated adversarial training. *arXiv preprint arXiv:2012.01791*, 2020.

# Appendix A

We show in Figure 2 an example of a zonotope being pushed through a single layer in a toy neural network to help build intuition for the reader into the ideas behind using zonotopes. In Figure 2 we start with two features which can each take values between 0 - 0.5. We initially convert this into a zonotope, so each feature now has a central term of 0.25, and an error term $\epsilon_1$ or $\epsilon_2$ with coefficients of 0.25. With the $\epsilon$ terms being able to take $\pm 1$ this draws the zonotope shape shown of a square.

This zonotope now gets pushed through the dense layer with weights

$$W = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}. \tag{4}$$

Multiplication of a zonotope by a constant affects all the terms, and we add zonotopes together term by term giving us $\hat{x}_3$ and $\hat{x}_4$. This gives the rotated rectangle shown in the middle plot. Note, that here we can see a advantage of the zonotope domain over just using intervals, as the rotated rectangle has a smaller area then just considering the upper and lower bounds of $\hat{x}_3$ and $\hat{x}_4$.

Finally, we apply a ReLU activation. As $\hat{x}_3$ has a positive upper bound and a lower bound of 0, the ReLU activation does not affect it. However, $\hat{x}_4$ has a negative lower bound and an upper bound

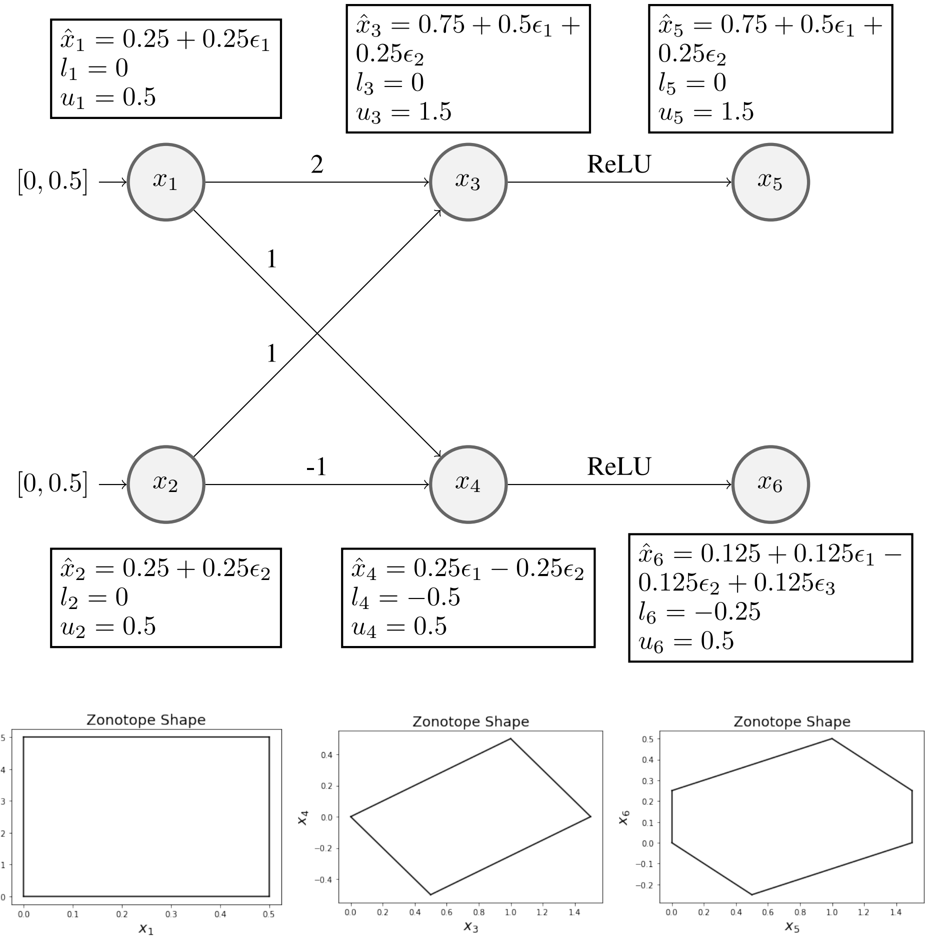
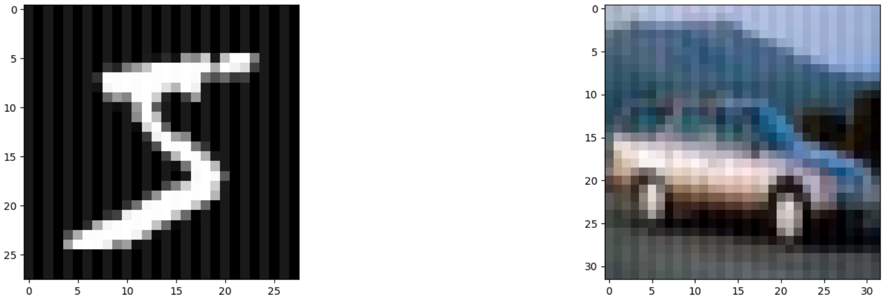

$\hat{x}_1 = 0.25 + 0.25\epsilon_1$
$l_1 = 0$
$u_1 = 0.5$
$\hat{x}_3 = 0.75 + 0.5\epsilon_1 + 0.25\epsilon_2$
$l_3 = 0$
$u_3 = 1.5$
$\hat{x}_5 = 0.75 + 0.5\epsilon_1 + 0.25\epsilon_2$
$l_5 = 0$
$u_5 = 1.5$
$[0, 0.5]$
$x_1$
2
$x_3$
ReLU
$x_5$
1
1
$[0, 0.5]$
$x_2$
-1
$x_4$
ReLU
$x_6$
$\hat{x}_2 = 0.25 + 0.25\epsilon_2$
$l_2 = 0$
$u_2 = 0.5$
$\hat{x}_4 = 0.25\epsilon_1 - 0.25\epsilon_2$
$l_4 = -0.5$
$u_4 = 0.5$
$\hat{x}_6 = 0.125 + 0.125\epsilon_1 - 0.125\epsilon_2 + 0.125\epsilon_3$
$l_6 = -0.25$
$u_6 = 0.5$
Zonotope Shape
$x_2$
$x_1$
Zonotope Shape
$x_4$
$x_3$
Zonotope Shape
$x_6$
$x_5$

Figure 2: Example of zonotopes being pushed through a toy layer, taking in 2 features and having a total of 4 weights. In each box we state the zonotope component as well as its upper $u$ and lower $l$ bounds.

above 0. Therefore, the application of a ReLU cannot be exactly captured- a concrete datapoint could either result in output of 0 from the ReLU or be unaffected. To include all the options in our analysis we use the DeepZono relaxation proposed in [18] and the application of it is shown for $\hat{x}_6$ (for the details of the calculation, interested readers can find the formulation in the paper [18]). The key take away is we now introduce a new error term $\epsilon_3$ which makes the analysis more imprecise. Further, the analysis is also now more complex as we need to track more error terms and this is reflected in the final shape being composed of 3 pairs of parallel lines.

## Appendix B

MNIST and CIFAR10 images with the backdoors used for the results in Table 1. We use $L_\infty$ bounds for the backdoor to follow the attacker model adversarial training is designed to protect against. In other words, if we allowed $L_0$ perturbations then backdooring to circumvent $L_\infty$ based adversarial training would not even be required as adversarial examples can be found which can evade the adversarial training. Hence, this represents perturbations which falls into what PGD should be able to protect against, and yet backdooring in this style can be used to introduce the hidden weaknesses desired by the attacker.



(a) Vertical stripes of magnitude $L_\infty = 0.1$. Model is adversarially trained to $L_\infty = 0.25$

(b) Vertical stripes of magnitude $L_\infty = 6/255$. Model is adversarially trained to $L_\infty = 8/255$

Figure 3: $L_\infty$ constrained backdoor keys.

## Appendix C

Here we have a brief overview of the functioning of defensive distillation and how it can be used as an attack. This was first proposed in the paper [27], and as it involves a few more steps compared to a backdoor style attack we detail the algorithm below.

**How do you train a network with defensive distillation?** In defensive distillation we train two neural networks: a teacher neural network and a student neural network. We apply a temperature scaling $T$ to the softmax of both networks. We first train the teacher neural network on data-label pairs $(x, y)$. After the teacher neural network has converged we use its predictions on the training dataset to re-label it. The student neural network still with temperature $T$ on the softmax is trained on the re-labeled dataset. Then, once converged, the student neural network's temperature is set back to 1. The defence is completed with the student neural network being the defended model.

**How does it protect against attack?** The core component is the temperature scaling factor which we apply during training, but remove at test time. At inference, this will case the inputs to the softmax function to be $T$ times larger. Due to this scaling, the output of the softmax will be close to a one hot vector. In fact, the components will be so close to either 0 or 1 that the 32-bit floating-point of non-predicted classes is rounded to 0 while the predicted class has a prediction of $\sim 1$. For similar reasons the the gradient, when represented as 32-bit floating-point, is rounded to 0 preventing the adversarial attack from making progress [7].

**How can an attacker circumvent it?:** Defensive distillation is an example of a *gradient masking* defence and so has a few shortcomings. Firstly it is vulnerable against black-box transfer attacks. However, stronger attacks can be performed by an attacker who knows the details of defensive distillation. Specifically, as was noted in [6] the key component of defensive distillation is the temperature scaling to cause vanishing gradients. An attacker can compute the adversarial examples by re-applying the temperature $T$ which re-scales the logits to appropriate values restoring gradient information enabling adversarial attack crafting. After crafting the adversarial examples the attacker then sends them to the original defender model with $T = 1$ and the adversarial examples function just as effectively.

**How can it be turned into an attack in FL?** In our scenario, we can use defensive distillation to achieve the goals of the attacker set out in Section 2.3. The model will look robust to a defender who evaluates it via standard application of the PGD attack. However, the attacker who knows that this is a defensively distilled model can craft their adversarial examples using temperature scaling as described above breaking the model.

## Appendix D

Neural network architectures used for our experiments. We use the following notation to describe our models: Conv$_{(s_w,s_h)}$ $C$ x $W$ x $H$ for a convolutional layer with $C$ output channels, and a kernel of width $W$ and height $H$. The stride size in width and height is given by $s_w$ and $s_h$ respectively. Additionally, FC $N$ indicates a fully connected layer with $N$ outputs.

MNIST model:

Conv$_{(2,2)}$ 16x4x4 → ReLU → Conv$_{(2,2)}$ 32x4x4 → ReLU → FC 1000 → ReLU → FC 10

CIFAR10 model:

Conv$_{(1,1)}$ 32x3x3 → ReLU → Conv$_{(2,2)}$ 64x4x4 → ReLU → Conv$_{(1,1)}$ 64x3x3 → ReLU → Conv$_{(2,2)}$ 128x4x4 → ReLU → FC 512 → ReLU → FC 512 → ReLU → FC 10