
FedRAD: Federated Robust Adaptive Distillation

Stefán Páll Sturluson
Department of Computing
Imperial College London, UK
stefan.sturluson20@imperial.ac.uk

Samuel Trew
Department of Computing
Imperial College London, UK
samuel.trew17@imperial.ac.uk

Luis Muñoz-González
Department of Computing
Imperial College London, UK
l.munoz-gonzalez@imperial.ac.uk

Matei Grama
Department of Computing
Imperial College London, UK
matei.grama17@imperial.ac.uk

Jonathan Passerat-Palmbach
Department of Computing
Imperial College London, UK
j.passerat-palmbach@imperial.ac.uk

Daniel Rueckert
Department of Computing
Imperial College London, UK
d.rueckert@imperial.ac.uk

Amir Alansary
Department of Computing
Imperial College London, UK
a.alansary14@imperial.ac.uk

Abstract

The robustness of federated learning (FL) is vital for the distributed training of an accurate global model that is shared among large number of clients. The collaborative learning framework by typically aggregating model updates is vulnerable to model poisoning attacks from adversarial clients. Since the shared information between the global server and participants are only limited to model parameters, it is challenging to detect bad model updates. Moreover, real-world datasets are usually heterogeneous and not independent and identically distributed (Non-IID) among participants, which makes the design of such robust FL pipeline more difficult. In this work, we propose a novel robust aggregation method, **Federated Robust Adaptive Distillation** (FedRAD), to detect adversaries and robustly aggregate local models based on properties of the median statistic, and then performing an adapted version of ensemble Knowledge Distillation. We run extensive experiments to evaluate the proposed method against recently published works. The results show that FedRAD outperforms all other aggregators in the presence of adversaries, as well as in heterogeneous data distributions.

1 Introduction

In recent years, deep learning models have shown themselves to be powerful prediction tools, often outperforming simpler models [1]. They have been applied in a wide variety of fields, such as healthcare [2], machine translation [3], and autonomous driving [4]. However, such methods are data hungry, and require large datasets to perform well [5]. In practice, data is usually expensive and time consuming to collect and annotate. Besides, privacy-sensitive data cannot be directly shared

globally, as it violates the current laws stated by regulatory bodies such as the general data protection regulations (GDPR)¹ and the health insurance portability and accountability act of (HIPAA)²

Federated Learning (FL) enables model training collaboratively in a decentralised manner without the need for data owners to hand over their data [6]. A typical FL pipeline consists of two main steps: (1) training a copy of the global model locally on the client’s private data, and (2) aggregating the local parameters into an updated global model. These two steps are repeated until converging to a final model. Model aggregation is usually performed by averaging the model updates (FedAvg). However, some of the participants could be faulty or malicious by sharing bad parameters’ updates, which can ruin the global model’s performance and prevent it from converging [7].

Recent works [8, 7, 9, 10, 11] have been published to mitigate the aforementioned limitations of FedAvg in the presence of malicious users and noisy adversaries. However, these methods fail to achieve the same good performance on heterogeneous data [12]. In practice, data heterogeneity is very common because of the different geographical locations or tools used to acquire the data. For example, the frequency of species of animals in natural image datasets collected in different continents will vary a lot, leading to clients from different continents to have label-skewed datasets.

In this paper, we propose a novel robust aggregation method that achieves state-of-the-art performance on heterogeneous client data, while still being able to detect adversaries or bad users and effectively defend against their attacks. The main contributions of this paper can be summarized as:

- A novel robust aggregator called Federated Robust Adaptive Knowledge Distillation (FedRAD), which utilizes a median-based scoring system along with knowledge distillation.
- Extensive evaluations of different aggregation methods with different types of attacks and levels of data heterogeneity.
- FedRAD outperformed all other methods in the presence of adversaries, both in IID and non-IID settings.
- The code is open source and publicly available, which can accelerate significantly the research in robust FL.

2 Background

Standard aggregation: Deep learning models are trained using stochastic gradient descent (SGD), where the parameters’ updates are computed on a mini-batch sampled from a local dataset, and back-propagated through the model. FedSGD [13] is a federated learning pipeline based on sampling a fraction of the gradients for every client, which are then sent to and averaged by a global server proportionally to the number of the samples on each client. Federated Averaging (FedAvg) [6] is a more efficient and generalized version of FedSGD by granting the clients to train their local model parameters on multiple batches of the local data before sharing the parameters of the model with the global server, instead of sharing the gradients as in FedSGD. Hence, FedAvg can be defined as the weighted average of the local updated models.

Model poisoning attacks: In distributed learning setup, the global model is aggregated from local models trained by honest participants. However, in a real world scenario, data may contain biases, such as wrongly annotated or missing labels or missing data. Such faulty clients sabotage the learning process by sending bad local models to the global server. In this work, we highlight two main strategies for bad clients, and assess their impact on the performance of the FL pipeline, namely:

- *Faulty or Byzantine:* Clients that add random noise to the model parameters before sending them back to the aggregator.
- *Malicious:* Clients that train their models in the same way as healthy clients, but instead of using correct labels they purposely train on incorrect labels, known as a label flipping attack.

Standard aggregation methods such as FedAvg are vulnerable to these bad local models.

Robust aggregation: Several works have been published to tackle the aforementioned drawbacks of FedAvg against adversaries. These methods rely mainly on the distribution of the local models or the

¹<https://gdpr-info.eu/>

²<https://www.hhs.gov/hipaa/index.html>

distance between local and global models. For example, COMED [7] utilises the coordinate-wise median of the client model parameters to construct the global model. Other methods, such as Krum, Multi-Krum [9], AFA [10] and FedMGDA+ [11] assigns a score to each client depending on the similarity of the models. These often consist of using a metric, such as cosine similarity or L2 distance between local and global model parameters. These scores are then used to determine which models are averaged and in what proportions. AFA and FedMGDA+ can also block clients permanently if they consistently are assigned a low score every federated round.

Data heterogeneity: In real-world scenario, participants of a distributed learning process are independent and have different distribution characteristics. The majority of robust aggregation methods are optimized to work on independently and identically distributed (IID) data between the clients. Although such methods can detect and block bad participants from the model aggregation step, benign clients with different data or labels than the majority will be also blocked and excluded during model aggregation. Inspired by Kairouz et al. [14], we adapt *label skewed* and *quantity skewed* data for the non-IID participants, simulated from IID datasets using the Dirichlet distribution.

Knowledge distillation: In this work, we utilize knowledge distillation for robust model aggregation. Hinton et al. [15] showed that soft predictions from a teacher model, or an ensemble of teacher models, can be used to train a student model. Recently, Lin et al. [16] proposed a method, FedDF, that uses Knowledge distillation in federated setting to learn a global model from an ensemble of client models using model fusion. A similar work by Chen and Chao [17], called FedBE, also uses knowledge distillation to teach a student model to match predictions from a Bayesian model ensemble. One drawback of FedDF and FedBE is that the aggregation server needs to have its own unlabelled dataset. However, this can be overcome using zero-shot learning, where synthetic data can be generated for this purpose [18, 19].

3 Federated robust adaptive distillation (FedRAD)

In this paper, knowledge distillation is performed using an ensemble of models to construct soft pseudolabels for an unlabeled dataset on the server side. A set of logits is predicted using the unlabeled data as an input of each model, from which the pseudolabels are constructed. FedBE applies the softmax function first to get the pseudo-probabilities from each model and then average those to make the pseudolabels, while FedDF instead averages the logits before applying a softmax function, resulting in soft pseudo-probabilities that can be used as labels during training. This works well when there are no attacks. However, when one or more classifiers are confidently predicting the incorrect label, it can affect the average of the logits, as shown in Table 3. Since logits outputs are unbounded, a single attacker can shift the average significantly. To tackle this drawback, we make use of the robust statistical characteristics of the median instead of the mean. This improves the robustness of the pseudolabels against outliers from the predicted models' logits.

The median logits pseudolabels are analysed empirically against attackers using the server-side unlabelled dataset. This can be done by counting how often each client's model has a logit output that is selected as the median value. Histograms of how often each client's model had the median output are shown in Figure 1. This is a 10-class prediction task on 10,000 images from the server-side dataset, so there total count is $10 \cdot 10,000 = 100,000$. The Faulty clients, i.e. ones that send back noisy updates, have a count close to 0. The malicious clients are on average chosen less often than the healthy clients, even in non-IID settings. This suggests that the pseudo-labels which are constructed using median logits will be less affected by faulty and malicious clients. By training the averaged model to match these pseudolabels, the effects of adversarial models is dampened.

Scoring system using medians: The median-based scoring algorithm assigns each client a score based on the frequency of which it outputs the median logit for a class prediction. This is done by counting how many times each client gave the median logit prediction, and then normalizing the counts such that the scores add up to 1, see Algorithm 1. These scores are then used for the weighted average of the client models. Since the median-based scores are on average lower for bad clients than healthy ones, this should decrease the effect of the bad model updates.

FedRAD: The aggregation server starts by initializing the weights of a global model. Then in each round, a subsample of clients are chosen and the global model is sent to them. Each client trains the global model using their data, then sends the model back to the aggregation server. Once the aggregation server has all the models, the unlabelled dataset is used to compute the median-based

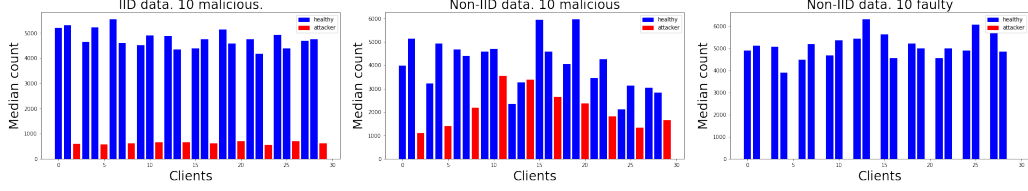


Figure 1: Median-counting histograms for prediction on 10,000 images from the MNIST dataset. There are 30 clients, 10 of which are attacking. Attackers are clients number 2, 5, 8, 11, 14, 17, 20, 23, 26, 29. Left: IID, 10 Malicious. Middle: Non-IID, 10 Malicious. Right: Non-IID, 10 Faulty.

Algorithm 1 Median-based scores (MedianScores)

Require: An ensemble $E = \{m_i\}_{i=1}^M$ of M models for C -dimensional classification problem. A dataset D (unlabeled).

Ensure: MedianScores(E, D):

```

 $P = \{0\}_{i=1}^M$  ▷ Counter
for each datapoint  $d \in D$  do
  for model  $m \in E$  do
     $\hat{y}_m \leftarrow m(d)$  ▷  $(1 \times C)$ -dimensional logits output of model  $m$  for input  $d$ .
  end for
   $med, idx \leftarrow median(\{\hat{y}_m\}_{m \in E}, dim = 0)$  ▷ Median in ensemble dimension
  for  $i$  in  $idx$  do
     $P_i \leftarrow P_i + 1$  ▷ Count how often model number  $i$  was the median
  end for
end for
 $P \leftarrow P / sum(P)$  ▷ Normalizing the scores such that  $\sum_i P_i = 1$ 
return  $P$ 

```

scores using Algorithm 1. These scores can be adjusted by the size of each client’s dataset. A weighted average is then taken of the client models to construct a new student model. Finally, the student model is trained using median-based knowledge distillation, and assigned to the new global model. These steps are repeated in federated rounds until convergence. Algorithm 2 demonstrates the pseudocode of the proposed FedRAD.

Algorithm 2 Federated Robust Adaptive Distillation (FedRAD)

Require: A set of participating clients $S = \{k_i\}_{i=1}^M$, each of which has a dataset of size n_k . Unlabelled server-side dataset D_s .

Ensure: FedRAD:

```

Initialise a global model with weights  $w_0^g$ .
for each round  $t = 1, \dots, T$  do
   $S_t \leftarrow$  Sample a subset of  $C$  fraction of clients from  $S$ 
  Send global model weights  $w_{t-1}^g$  to clients in  $S_t$ 
  for each client  $k \in S_t$  in parallel do
     $w_t^k \leftarrow$  Updated global model using client data
  end for
   $p_t \leftarrow MedianScores(\{w_t^k\}_{k \in S_t}, D_s)$  ▷ See algorithm 1
   $p_t \leftarrow \left\{ \frac{n_k \cdot p_t^k}{\sum_k n_k \cdot p_t^k} \right\}_{k \in S_t}$  ▷ Adjust by size of datasets
   $w_t^g \leftarrow \sum_{k \in S_t} p_t^k \cdot w_t^k$  ▷ Element-wise weighted average
   $w_t^g \leftarrow KnowledgeDistillation(w_t^g, \{w_t^k\}_{k \in S_t}, D_s)$  ▷ Median-based KD
end for

```

4 Experiments and results

Simulating non-IID data: Here, we use the Dirichlet distribution which is a multivariate continuous probability distribution. It is parameterized by a K -dimensional vector $\alpha = \{\alpha_i\}_{i=1}^K, \forall i : \alpha_i > 0$. Its Probability Density Function (PDF) is

$$f(\mathbf{X}; \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i-1}$$

where $B(\alpha)$ is a normalizing constant and $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ is the gamma function.

Samples from the Dirichlet distributions are K -dimensional vectors. The support is the open standard $(K - 1)$ -simplex, $\{x \in \mathbb{R}^K : x_i \in (0, 1) \text{ for all } i = 1, 2, \dots, K; \sum_{i=1}^K x_i = 1\}$.

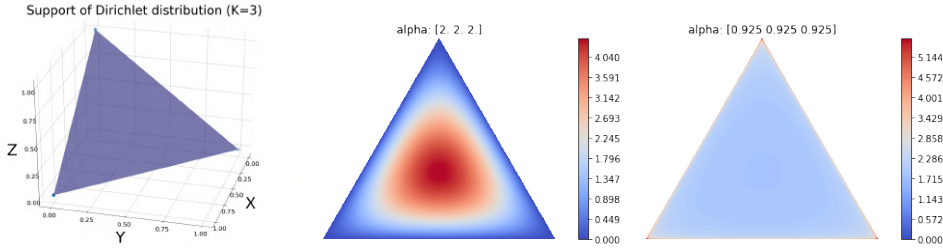


Figure 2: Support and Contour maps of PDFs of Dirichlet distributions ($K=3$). Left: Support. Middle: Symmetrical $\alpha > 1$. Right: Symmetrical $\alpha < 1$.

A symmetrical Dirichlet distribution can be used to simulate a non-IID split of datasets between agents. This is done by distributing the samples in each class unequally between agents, in proportions sampled from a symmetrical Dirichlet distribution. If we have a dataset with C classes and want to divide the data among K different agents, we can define a symmetrical K -dimensional Dirichlet distribution $Dir(\alpha)$ where $\alpha = \alpha \cdot \mathbf{1} = \{\alpha\}_{i=1}^K$. We then sample C vectors from the distribution, one for each class, and divide the data in each class between the agents in proportion with the sampled values. The α hyperparameter can be tuned to control how unequally the data classes are divided between agents. A high value of α leads to a low variance in proportion, resulting in more equal splits between agents, whereas a low α increases the variance of the proportions, leading to a more non-IID data split.

Implementation: In order to thoroughly assess the performance of FedRAD, we have implemented several aggregation methods and two types of adversarial attacks. *Faulty* clients are simulated by adding Gaussian noise with a variance of 20 to the model parameters. *Malicious* clients are simulated by setting all their data labels to 0 before training.

Non-IID data distributions are simulated with three levels of heterogeneity: IID data split with slight quantity skews, a slightly non-IID split sampled from a symmetrical Dirichlet distribution with $\alpha = 0.5$ and a very non-IID split which uses $\alpha = 0.1$, see Figure 3

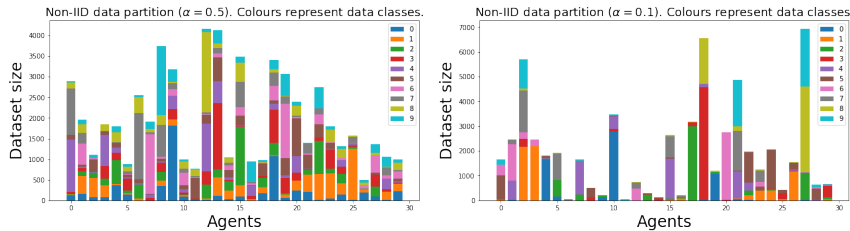


Figure 3: Examples of simulated non-IID data splits using samples from a symmetrical Dirichlet distribution. Left: $\alpha = 0.5$. Right: Symmetrical $\alpha = 0.1$.

A Knowledge distillation module is implemented by training a student model to emulate predictions from an ensemble of models. Two aggregation methods which use Knowledge Distillation, FedDF [16] and FedBE [17], are also implemented.

For our experiments we use the MNIST dataset [20], which is a labeled image dataset of handwritten digits from 0 to 9. It has a total of 70 thousand images of size 28×28 . The training set includes 60 thousand images, and the test set has the remaining 10 thousand. There are roughly the same number of images in each class. Our classifier consists of a 3-layer fully-connected neural network that outputs the logits it assigns to each of the 10 classes. A Softmax function is then used to get the predicted probabilities. We ran our experiments for 30 communication rounds, with 5 local epochs and a local learning rate of 0.05. Performance is measured as the Error rate calculated on the test set.

FedRAD code and implementation is publicly available on GitHub³

Results: The novel aggregators proposed in this project, median-based FedDF (FedDFmed) and FedRAD, outperformed all other aggregators in the presence of attacks, both in IID and non-IID settings. Using medians instead of average logits for knowledge distillation is effective to combat malicious label-flipping attacks. Furthermore, the median-based scoring system enhances the robustness of the trained models by detecting faulty attackers and assigning them very low scores. It is also successful in giving lower scores to malicious attackers than healthy agents. FedRAD combines the novel median-based knowledge distillation with the novel median-based scoring system.

Table 1 shows that it is more difficult to detect malicious attacks than faulty ones by examining the model parameters, as malicious agents use the same training procedure as healthy clients and aren't extremely noisy like the faulty models. The model weights of malicious agents are similarly distributed to healthy ones, and in non-IID settings it becomes difficult to detect malicious agents using similarity metrics. Table 1 also demonstrates that FedRAD under a variety of attacks and data heterogeneity outperforms all the other aggregation methods.

Figure 4 displays the learning curves in experiments with both faulty and malicious attackers in IID and non-IID settings. With 5 faulty and 5 malicious attacks, COMED did well in IID scenarios, but as usual failed for non-IID scenarios. However, FedRAD consistently performed well under both types of attacks at the same time. More results can be found in Appendix D.

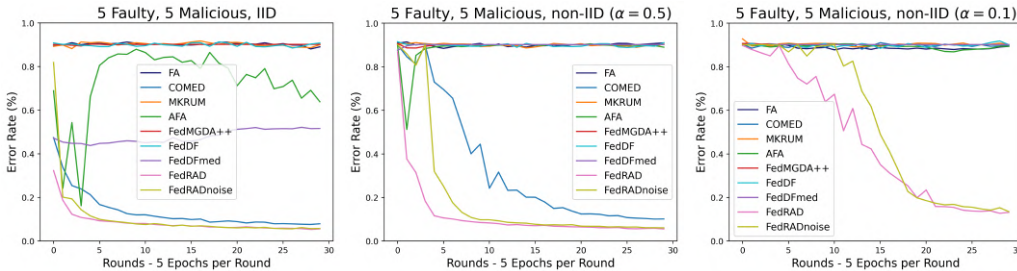


Figure 4: Effects of non-IID data with 5 faulty and 5 malicious attackers.

5 Conclusions

In this work, we propose FedRAD for robust federated aggregation using a scoring system based on the median statistic and network distillation. In contrary to previous robust aggregation methods that use the high-dimensional vectors of parameters of the models, our method use the output predictions from the models for detecting adversaries in the presence of heterogeneous client data.

Extensive experiments were conducted using the MNIST dataset to evaluate the performance of different baseline methods in the presence of attackers, for both IID and non-IID client data. Existing aggregators, like COMED and Multi-Krum, were shown to break down when client data is heterogeneous. Existing adaptive aggregators, AFA and FedMGDA+, were also shown to be unable to detect and block adversaries in non-IID settings.

To conclude, FedRAD outperforms all other aggregation methods in the presence of attacks, and is the only aggregator to successfully combat adversaries in both IID and non-IID settings. We will investigate the performance of FedRAD with privacy preserving techniques such as differential privacy in the future.

³<https://github.com/stefansturlu/FederatedMedical>

Attacks	Aggregator	Error rates (%)		
		IID	non-IID	
			$\alpha = 0.5$	$\alpha = 0.1$
No attacks	FA	5.24 ± 0.07	5.68 ± 0.11	9.13 ± 0.82
	COMED	5.30 ± 0.08	6.38 ± 0.18	81.15 ± 12.94
	MKRUM	5.68 ± 0.16	7.63 ± 0.62	40.93 ± 13.51
	AFA	5.28 ± 0.17	5.94 ± 0.16	10.36 ± 0.96
	FedMGDA++	6.28 ± 0.25	12.15 ± 5.39	21.12 ± 3.84
	FedDF	5.43 ± 0.10	5.82 ± 0.15	9.17 ± 0.40
	FedDFmed	5.45 ± 0.08	5.86 ± 0.25	9.24 ± 0.43
FedRAD	5.29 ± 0.16	5.82 ± 0.15	9.38 ± 0.57	
10 Faulty	FA	90.03 ± 0.96	89.58 ± 0.50	90.10 ± 0.63
	COMED	5.39 ± 0.14	6.44 ± 0.28	69.22 ± 22.75
	MKRUM	89.66 ± 0.63	90.37 ± 0.37	90.03 ± 0.65
	AFA	70.97 ± 32.56	90.12 ± 0.47	89.70 ± 0.33
	FedMGDA++	89.82 ± 0.71	89.37 ± 0.28	89.89 ± 0.43
	FedDF	90.06 ± 0.82	89.65 ± 1.01	89.71 ± 0.62
	FedDFmed	81.02 ± 3.56	90.14 ± 0.54	90.11 ± 0.61
FedRAD	5.44 ± 0.07	5.91 ± 0.31	12.63 ± 3.65	
10 Malicious	FA	17.33 ± 3.64	74.87 ± 30.66	38.09 ± 27.79
	COMED	25.11 ± 1.39	48.51 ± 21.52	90.08 ± 0.40
	MKRUM	31.15 ± 25.15	10.82 ± 2.19	60.78 ± 15.24
	AFA	17.68 ± 4.46	90.20 ± 0.00	90.20 ± 0.00
	FedMGDA++	90.20 ± 0.00	90.20 ± 0.00	90.20 ± 0.00
	FedDF	65.20 ± 28.69	76.12 ± 28.16	61.49 ± 31.78
	FedDFmed	8.34 ± 0.25	9.24 ± 1.23	16.74 ± 3.87
FedRAD	5.89 ± 0.20	6.55 ± 0.43	13.09 ± 1.97	
5 Faulty, 5 Malicious	FA	89.64 ± 0.55	89.75 ± 0.75	89.62 ± 0.63
	COMED	8.00 ± 0.42	10.21 ± 0.93	89.10 ± 0.90
	MKRUM	89.60 ± 0.36	89.66 ± 0.43	89.82 ± 0.67
	AFA	67.62 ± 6.64	67.82 ± 30.59	80.97 ± 17.03
	FedMGDA++	90.20 ± 0.00	90.21 ± 0.00	89.31 ± 1.61
	FedDF	90.39 ± 0.57	90.17 ± 0.67	89.97 ± 0.18
	FedDFmed	55.11 ± 5.41	90.20 ± 0.00	90.20 ± 0.00
FedRAD	5.63 ± 0.20	6.15 ± 0.40	10.95 ± 1.22	

Table 1: Test set error rate for MNIST after 30 rounds. Average and standard deviation of test-set error rates shown for 5 different random seeds. The best performing methods in each category, and those not statistically significantly different according to a t-test, are in bold.

Acknowledgments and Disclosure of Funding

Acknowledgements This work is supported by the UK Research and Innovation London Medical Imaging and Artificial Intelligence Centre for Value Based Healthcare, and in part by the Intramural Research Program of the National Institutes of Health Clinical Center.

References

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [2] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and vascular neurology*, 2(4), 2017.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [4] Alex Bewley, Jessica Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, and Alex Kendall. Learning to drive from simulation without real world labels. In *2019 International conference on robotics and automation (ICRA)*, pages 4818–4824. IEEE, 2019.
- [5] Ziad Obermeyer and Ezekiel J Emanuel. Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216, 2016.
- [6] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.
- [7] 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.
- [8] Matei Grama, Maria Musat, Luis Muñoz-González, Jonathan Passerat-Palmbach, Daniel Rueckert, and Amir Alansary. Robust aggregation for adaptive privacy preserving federated learning in healthcare. *arXiv preprint arXiv:2009.08294*, 2020.
- [9] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.
- [10] 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.
- [11] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Fedmgda+: Federated learning meets multi-objective optimization. *arXiv preprint arXiv:2006.11489*, 2020.
- [12] 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.
- [13] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.
- [14] 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.
- [15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [16] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *arXiv preprint arXiv:2006.07242*, 2020.
- [17] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2, 2020.
- [18] Paul Micaelli and Amos Storkey. Zero-shot knowledge transfer via adversarial belief matching. *arXiv preprint arXiv:1905.09768*, 2019.
- [19] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, pages 4743–4751. PMLR, 2019.
- [20] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.