A Systematic Approach to Robustness Modelling

Charles Meyers[1]

[1]Department of Computing Science, Umeå University, Umeå, Sweden

***Abstract*—Considering the growing prominence of production-level AI and the threat of adversarial attacks that can poison a machine learning model against a certain label, evade classification, or reveal sensitive data about the model and training data to an attacker, adversaries pose fundamental problems to machine learning systems. Furthermore, much research has focused on the inverse relationship between robustness and accuracy, raising problems for real-time and safety-critical systems particularly since they are governed by legal constraints in which software changes must be explainable and every change must be thoroughly tested. While many defenses have been proposed, they are often computationally expensive and tend to reduce model accuracy. I have been evaluating a large suite of attacks and defenses and developing a framework for analyzing any machine-learning system from a safety-critical perspective using adversarial noise to find the upper bound of the failure rate. By developing systematic approaches to robustness testing, I hope to develop real-time tests that guarantee performance beyond the current test/train split paradigm.**

I. On the use Adversarial Attacks for AI Pipeline Verification

Artificial Intelligence (AI) pipelines are often long-running and complex software tool-chains with many tunable hyper-parameters. Managing, tracking, and controlling for various parameters is non trivial, but many management tools are available [1], [2], [3]. In general, a dataset is split into **training** and **test** sets. The training set is then used to determine the best configuration of a given model architecture on a given hardware architecture with the expectation that it will generalize both on the withheld test set and on new data generated by users via application programming interface (API) calls. To verify the training process, the test set validated against the *inference* configuration of a model which may run on different hardware than the **training** configuration to reduce cost, latency, or power consumption.

A. Adversarial Attacks

In the context of machine learning, an adversarial attack refers to deliberate and malicious attempts to manipulate or exploit machine learning models. Adversarial attacks are designed to deceive or mislead the model's behavior by introducing carefully crafted input data that can cause the model to make incorrect predictions or produce undesired outputs.

The goal of an adversarial attack is often to exploit vulnerabilities in the model's decision-making process or to probe its weaknesses. These attacks can occur during various stages of the machine learning pipeline, including during training, inference, or deployment.

- Evasion Attacks: These attacks aim to manipulate input data during the inference phase to deceive the model into misclassifying or ignoring certain inputs. Attackers carefully craft perturbations or modify the input features to mislead the model while still appearing similar to the original input [4], [5], [6], [7], [8].
- Poisoning Attacks: In poisoning attacks, the attacker intentionally injects malicious or manipulated training samples into the training dataset. The goal is to influence the model's behavior during training so that it learns to make incorrect predictions or exhibit unwanted behaviors when presented with specific inputs [9], [10].
- Inference Attacks: These attacks exploit the model's output or responses to obtain sensitive information about the training data or other confidential details. By observing the model's predictions or confidence scores for carefully crafted inputs, attackers can extract information that should ideally be kept private [11], [12].
- Model Inversion Attacks: Model inversion attacks aim to infer sensitive information about the training data or proprietary model by exploiting the model's outputs. Attackers utilize the model's responses to iteratively

reconstruct or approximate training examples that are similar to the ones used during training [11], [13], [14].

B. Security

By using various types of attacks, we can model the security of AI pipelines as well. In order to measure robustness of a given model, it is assumed that the attacker knows most things about the model, including the distribution, shape, and feature space of the training set; the type of model used and it's parameter space; the gradients with respect to the optimization criteria; and feedback from the model in the form of model probability output. While it may seem like a prohibitively large set of assumptions, I outline possible attack vectors below. In our case, the attacker queries the model with an adversarial sample and is given the ℓ_∞ norm which returns the largest deviation of a single feature for a given sample rather than the more granular information provided by other standard distance metrics, like the ℓ_2 or ℓ_0 norms. It also ensures that no single feature is perturbed by more than d_{max}.

a) Perfect Knowledge: : While assumed the adversary has access to the model gradients with respect to the loss function, it can be approximated through Monte Carlo methods or via other attacks [15], [16]. It is not necessary to know all of the model parameters, just the weights and biases that compose the fitted model. Although even this constraint is broken by other attacks [15], [16]. The adversary can transform sample data, but must remain within a maximum distance d_{max} for each feature. Other works [17], [18], [19] try to minimize the requisite perturbation distance. In others cases, perfect knowledge is provided normally by the peer-review process and published model weights. However, many models are proprietary and can only be accessed through an API that returns only the classification, either as a probability distribution or the $argmax$ of that distribution. Attacks with perfect knowledge are considered to be **black-box attacks** [11].

C. Privacy

Even though our attack scenario only includes perfect knowledge, prior research [20], [21], [16], [15], [22] has shown that a surrogate model and data-set can be used to approximate $f(x)$ by $\hat{f}(x)$ and build a model using the class labels provided by the model at test-time. Tramèr et al. [23] examined popular machine learning as a service platforms that return confidence values as well as class labels, showing that an attacker can build a proxy model by querying $p+1$ random $p-$dimensional inputs for unknown $p+1$ parameters. Further research [20] were able to reverse engineer the training data-set through black-box attacks against a model that returns confidence levels, with the caveat that the inferred data might be a meta-prototypical example that does not appear in the original data-set. Fortunately for our attacker, such examples are still useful for determining the underlying data distributions even if they manage to preserve some of the privacy of the original data-set. Shokri et al. [24] presented a membership inference attack that determines whether a given data point belongs to the same distribution as the original training data using a set of proxy models. There are myriad ways for an attacker to get access to otherwise private data using nothing but standard machine learning APIs. Attacks that only require access to these APIs are considered **white-box attacks**.

D. Safety

The ISO standards [25] define the Safety Inegrity Level (SIL) in failures/per hour, which I have converted to failures per second in Table I. If assume that *accidental* adversarial errors are possible in real-world systems due to things like dust, lens flare, component failure, packet loss, *etc.*, it naturally follows that the adversarial failure rate is an estimate of the models behavior at the edge or in the 'worst-case scenario'. That is, the *adversarial failure rate* is an estimate of the upper bound of the real-world failure rate in adverse but otherwise mundane circumstances. However, due to the large number of samples required by regulatory standards and the strenuous testing requirements of safety-critical software these evaluations become an infeasible way to verify that a model only fails once across the required number of samples (see Table I), especially if we would like to be highly confident of that estimation.

TABLE I
ACCEPTABLE FAILURE RATES FOR DIFFERENT SIL LEVELS IN WHICH A SINGLE DEATH IS POSSIBLE, MEASURED IN FAILURES PER SECOND.

SIL	On-demand Operation	Continuous Operation
I	$[10^{-6}, 10^{-5})$	$[10^{-10}, 10^{-9})$
II	$[10^{-7}, 10^{-6})$	$[10^{-11}, 10^{-10})$
III	$[10^{-8}, 10^{-7})$	$[10^{-12}, 10^{-11})$
IV	$[10^{-9}, 10^{-8})$	$[10^{-13}, 10^{-12})$

E. Robustness

Robustness, then, is a measure of how well a model resists these *induced* failures. For example, we could see how different AI-pipelines influence the accuracy given model architecture and dataset using the *Percent Change in Accuracy* ($\%\Delta ACC$):

$$\%\Delta\text{ACC} = \frac{\text{Acc.} - \text{Control Acc}}{\text{Control Acc}} \cdot 100 \tag{1}$$

where Acc refers to the *accuracy* and $Control$ refers to the performance of the unchanged model on the benign dataset. This measures the marginal risk of failure for a particular model change (defense) in the adversarial case when compared to the benign case. We could extend this to any other success metric like loss, the number of queries it takes to steal a database entry, or the time it takes to steal a model (see: Section I-C for more examples). In general [26], we can examine *Relative Change in Failure Rate ($\Delta\eta$):*

$$\Delta\eta = \frac{\eta_{\text{control}} - \eta}{\eta} \tag{2}$$

where η refers to the failure rate, $Control$ refers to the unchanged model. Taken together, these two metrics allow us to measure the marginal risk of a given defense in both the benign and adversarial circumstances. In both cases, a positive number indicates an improvement in relative risk and a negative number indicates a worsening of relative risk, Eq. 1 in the context of accuracy and Eq. 2 in the context of failure rate.

II. Adversarial Attacks for CI/CD

Historically, marginal gains in model performance have relied on exponentially larger models to produce increasingly marginal gains [27]. These models rely on increasingly larger datasets [27], [28], [29], which increasingly come from fewer sources [30], leading to gender-biased models [31], racism [32], and fatal design errors [33] This is a trend that goes back decades [34], [35], [32], leading to, for example, significantly higher fatality in car accidents for female-bodied people [36] or neural networks that unintentionally encode racial information from medical imaging data alone [37]. Furthermore, data collection can be expensive [38], raises serious privacy concerns [39], increases time to market [40], and impedes development speed [41]. Furthermore, research is focused on metrics that tend to be optimistic at best [42]. As noted by many researchers [42], [5], [43], [26], test/train split optimization can only find failures from 'in-distribution' data. Instead, we need strong guarantees that our models will not fail subject to the aforementioned standards. It is also critical that these test run fast enough that they can be incorporated into the model-development feedback loop and keep, for example, untested AI software from needing an actual car to do verification test. Failure rate analysis has been widely explored in other fields [44], but there's very little published research in the context of machine learning models.

A. Quality Assurance

The exponential distribution is a probability distribution that models the time until an event occurs in a continuous-time setting. In the context of failure rate, the exponential distribution is often used to characterize the failure or survival time of a system or component. The probability distribution function is given by:

$$f(x) = \begin{cases} \eta e^{-\eta x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The mean or expected value of the exponential distribution is given by:

$$\mu = \frac{1}{\eta}$$

The variance of the exponential distribution is given by:

$$\sigma^2 = \frac{1}{\eta^2}$$

This distribution is often used to model the case where a failure rate is invariant across samples [44]. However, we can also measure the effects of covariates and do this in an accelerated way if we are clever enough.

III. Accelerated Failure Rate Models

Accelerated failure rate models are statistical models used to analyze multivariate effects on the observed failure rate and to generate a model that's used to predict the failure rate across a wide variety of circumstances. The advantage

of using Weibull, Log-Logistic, or Log-Normal distributions over methods like Kaplan-Maier [44] is that the latter is non-parametric, meaning that the resulting model does little to explain the model's performance across the various configurations. Additionally, semi-parametric models like the Cox Proportional Hazard model assume that the failure rate is constant over the covariates (usually time). Subsequently, this would do a poor job of modelling the effect of an attack, defence, or model configuration that changes the run-time or the prediction accuracy relative to the control. Furthermore, there is substantial empirical evidence [45] that the efficacy of at least some attacks increases as we change various attack parameters. Therefore, in order to generate an explainable model that encapsulates the effect of the covariates, we can use a distribution like Weibull, Log-Logistic, or Log-Normal [44].

A. Optimization

Because of the relatively small run-time requirements of this approach (when compared to testing against massive in-distribution test sets), this method could, for example, act as a unit test in machine learning applications rather than relying on full-system integration tests to evaluate changes to a single model, signal processing technique, data storage format, or API access mechanism. It could also be used to highlight error-prone classes or other subsets of data to reduce error or create synthetic samples. Furthermore, by isolating changes and testing them as quickly as possible, it's much easier to parse cause and effect when compared to full-system integration tests that could include many changes from many different development teams and require live and potentially dangerous systems (like cars or MRI machines) to effectively test. To further increase development velocity, metrics Eq. 1 and Eq. 2 have been proposed [26] as standards for evaluating not only the efficacy of a given change, but as tools to quantify the marginal risk associated with each change, as dictated by the ISO 26262 standard [25].

IV. Future Trends

As stated by many researchers [42], [5], [43], [26], there is a reliability crisis effecting Artificial Intelligence. A very recent paper [46] recently evaluated around 20 thousand jupyter notebooks that have been cited in medical studies. However, they found that fewer that 800 of them could run due to dependency issues. Interestingly, they also show how this trend has been improving over time. Whether that's due to time or an actually improved code-base is left to future research.

A. Lower-power Hardware

Since the continued existence and efficacy of these attacks raises questions about the ability of these architectures to generalize since things like bit-depth [47], training-noise [48], label-noise [49], and image resolution [45] greatly vary the failure rate. In the real-world, effective resolution will change between individuals (e.g. a medical scan) or while moving (e.g. an autonomous vehicle). Even random noise drawn from approximately the same distribution as the training set[48], [49] increases the benign failure rate by an order of magnitude or two. However, some research [26] suggests that using low-bit-depth hardware could lower power requirements while *increasing robustness*, though proving this is left to future research.

B. Edge computing and Federation

Edge computing [50], [51] inverts the normal paradigm for datacenters. Instead of having on-demand, always available, high-bandwidth, and redundant systems, we rely on faulty, unreliable, low-power systems with variable bandwidth on a distributed network. The benefits for AI applications could be immense. Drones and autonomous vehicles could offload their processing, content distribution networks can move from giant, remote datacenters to a bandwidth-minimizing swarm dotted around the neighborhood. Medical care could be cheaper, faster, and more accurate. However, we have a long way to go. Fortunately, model federation [52] solves many of these problems. It could solve privacy concerns for many applications [53], [54].

References

[1] dvc.org, "Dvc- data version control," Github, 2023. [Online]. Available: https://github.com/iterative/dvc.org

[2] O. Yadan, "Hydra - a framework for elegantly configuring complex applications," Github, 2019. [Online]. Available: https://github.com/facebookresearch/hydra

[3] Kubernetes, "Kubernetes–an open source system for managing containerized applications," Github, June 2019. [Online]. Available: https://github.com/kubernetes/kubernetes

[4] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, "Evasion Attacks against Machine Learning at Test Time," *arXiv:1708.06131 [cs]*, vol. 7908, pp. 387–402, 2013. [Online]. Available: http://arxiv.org/abs/1708.06131

[5] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," *arXiv:1608.04644 [cs]*, Mar. 2017. [Online]. Available: http://arxiv.org/abs/1608.04644

[6] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv:1712.09665*, 2017.

[7] S. Kotyan and D. Vasconcellos Vargas, "Adversarial robustness assessment: Why both l_0 and l_∞ attacks are necessary," *arXiv e-prints*, pp. arXiv–1906, 2019.

[8] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," in *IEEE symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.

[9] B. Biggio, B. Nelson, and P. Laskov, "Poisoning Attacks against Support Vector Machines," *arXiv:1206.6389 [cs, stat]*, Mar. 2013. [Online]. Available: http://arxiv.org/abs/1206.6389

[10] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 957–11 965.

[11] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv:1810.00069 [cs, stat]*, 2018.

[12] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4954–4963.

[13] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *International conference on machine learning*. PMLR, 2021, pp. 1964–1974.

[14] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 880–895.

[15] X. Wang, J. Li, X. Kuang, Y.-a. Tan, and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.

[16] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv:1810.00069*, 2018.

[17] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *International Conference on Machine Learning*, 2012.

[18] D. Stutz, M. Hein, and B. Schiele, "Confidence-calibrated adversarial training: Towards robust models generalizing beyond the attack used during training," *International Conference on Machine Learning*, 2019.

[19] B. Li, Y. Vorobeychik, and X. Chen, "A general retraining framework for scalable adversarial classification," *Workshop on Adversarial Training, Neural Information Processing Systems*, 2016.

[20] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[21] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[22] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, vol. 10, no. 3, pp. 137–150, 2015.

[23] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th {USENIX} Security Symposium Security 16)*, 2016, pp. 601–618.

[24] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[25] I. S. Organization, "ISO 26262-1:2011, road vehicles — functional safety," https://www.iso.org/standard/43464.html (visited 2022-04-20), 2018.

[26] C. Meyers, T. Löfstedt, and E. Elmroth, "Safety-critical computer vision: An empirical survey of adversarial evasion attacks and defenses on computer vision systems," *Artificial Intelligence Review*, 2023.

[27] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Compute and energy consumption trends in deep learning inference," *arXiv:2109.05472*, 2021.

[28] V. Vapnik, E. Levin, and Y. Le Cun, "Measuring the vc-dimension of a learning machine," *Neural computation*, vol. 6, no. 5, pp. 851–876, 1994.

[29] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, "Learnability and the vapnik-chervonenkis dimension," *Journal of the ACM*, vol. 36, no. 4, pp. 929–965, 1989.

[30] B. Koch, E. Denton, A. Hanna, and J. G. Foster, "Reduced, reused and recycled: The life of a dataset in machine learning research," *arXiv preprint arXiv:2112.01716*, 2021.

[31] K. Lu, P. Mardziel, F. Wu, P. Amancharla, and A. Datta, "Gender bias in neural natural language processing," *Logic, Language, and Security: Essays Dedicated to Andre Scedrov on the Occasion of His 65th Birthday*, pp. 189–202, 2020.

[32] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*. PMLR, 2018, pp. 77–91.

[33] V. A. Banks, K. L. Plant, and N. A. Stanton, "Driver error or designer error: Using the perceptual cycle model to explore the circumstances surrounding the fatal tesla crash on 7th may 2016," *Safety science*, vol. 108, pp. 278–285, 2018.

[34] W. A. Corsaro, "Something old and something new: The importance of prior ethnography in the collection and analysis of audiovisual data," *Sociological Methods & Research*, vol. 11, no. 2, pp. 145–166, 1982.

[35] D. Ramirez, J. McDevitt, and A. Farrell, *A resource guide on racial profiling data collection systems: Promising practices and lessons learned*. US Department of Justice, 2000.

[36] L. Evans and P. H. Gerrish, "Gender and age influence on fatality risk from the same physical impact determined using two-car crashes," *SAE transactions*, pp. 1336–1341, 2001.

[37] J. W. Gichoya, I. Banerjee, A. R. Bhimireddy, J. L. Burns, L. A. Celi, L.-C. Chen, R. Correa, N. Dullerud, M. Ghassemi, S.-C. Huang *et al.*, "Ai recognition of patient race in medical imaging: a modelling study," *The Lancet Digital Health*, vol. 4, no. 6, pp. e406–e414, 2022.

[38] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.

[39] C. Bloom, J. Tan, J. Ramjohn, and L. Bauer, "Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles," in *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[40] H. Lam, "New design-to-test software strategies accelerate time-to-market," in *IEEE/CPMT/SEMI 29th International Electronics Manufac-*

turing Technology Symposium (IEEE Cat. No. 04CH37585). IEEE, 2004, pp. 140–143.

[41] B. J. Zirger and J. L. Hartley, "The effect of acceleration techniques on product development time," *IEEE Transactions on Engineering Management*, vol. 43, no. 2, pp. 143–152, 1996.

[42] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Machine Learning*, 2017.

[43] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," *arXiv:2003.01690 [cs, stat]*, Aug. 2020. [Online]. Available: http://arxiv.org/abs/2003.01690

[44] M. J. Bradburn, T. G. Clark, S. B. Love, and D. G. Altman, "Survival analysis part ii: multivariate data analysis–an introduction to concepts and methods," *British journal of cancer*, vol. 89, no. 3, pp. 431–436, 2003.

[45] E. Meyers, Löfstedt, "Safety-critical computer vision: An empirical survey of adversarial evasion attacks and defenses on computer vision systems," *Springer Artificial Intelligence Review*, 2023.

[46] S. Samuel and D. Mietchen, "Computational reproducibility of jupyter notebooks from biomedical publications," *arXiv preprint arXiv:2308.07333*, 2023.

[47] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv:1704.01155*, 2017.

[48] V. Zantedeschi, M.-I. Nicolae, and A. Rawat, "Efficient defenses against adversarial attacks," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 39–49. [Online]. Available: https://doi.org/10.1145/3128572.3140449

[49] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 656–672.

[50] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, 2019.

[51] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[52] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[53] R. Shamim, M. Arshad, and V. Pandey, "A machine learning model to protect privacy using federal learning with homomorphy encryption."

[54] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*. PMLR, 2019, pp. 634–643.