DS603: Robust Machine Learning **Autumn 2025**

Lecture 1 – 30/07/2025

Lecturer: Arjun Bhagoji *Scribe: Rahul Kumar Yadav*

1.1 What is Machine Learning?

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.", as given in the book Machine Learning by Tom M. Mitchell.

1.1.1 Some Machine Learning Uses in Modern Times-

1. Using AI for surveillance.
2. Content Moderation using AI.
3. In cybersecurity(Darktrace uses AI to respond to hacks accordingly).
4. Financial Transactions(MasterCard using AI to improve the approval rate of genuine transactions)

1.2 Safety Risks(Machine Learning is prone to failure)

Machine Learning models have been seen to have various problematic features, depending on how they have been trained. Some of them are:-

1. Making unfair decisions.
2. They can leak private data.
3. Training of deep learning models, especially the modern LLM ones consume humongous amount of compute and electricity, resulting in large amount of carbon emmissions.
4. They can be fooled and thus can be taken advantage of. For e.g.:- Using LLMs to get information on how to create explosives, etc.

1.3 General idea behind adversarial examples

Losely speaking, examples or samples are said to be adversarial(whose main goal is to fool the machine learning model), by making the model to misclassify the example into something which it isn't according to human perception. For e.g.: Dogs being classified as cats by just inducing a small perturbation in the image of dog, which is largely not visible to humans, and humans are not susceptible to these.

A general way of creating the adversarial samples is to perturb the samples in such a manner, which can change the label predicted by the machine learning model associated with the original samples. There are various kinds of attacks associated with how to generate adversarial samples for e.g.: Fast Gradient Sign Method(FGSM), Projected Gradient Descent(PGD), are some of the common ones.

To counteract it or to make our models robust, we do adversarial training in which we make our model learn those perturbations by keeping the labels same as the original one, i.e. if our original sample and labels were (x, y), then in the case of adversarial learning our samples become the perturbed samples we get from the original samples, while keeping the labels same. $(x + \delta, y)$. More formal discussions would follow in the upcoming lectures.

1.4 Robustness, Privacy, and Fairness: A Primer

We want to:

- **Understand** when and how ML models fail
- **Reason** about rigorous limits on robustness, privacy, and fairness
- **Build** more reliable ML systems

Robustness

- *Anomalies & noise:* How do models behave under out-of-distribution inputs or corrupted labels?
- *Training-time attacks:* Poisoning a small fraction of the training set to induce large errors at test time.
- *Test-time (adversarial) attacks:* Adding tiny, humanely imperceptible perturbations to inputs that lead to misclassifications.
- *Defenses:* Adversarial training, robust statistics, data sanitization.

Privacy

- *Membership inference:* Can an adversary tell if a given example was in the training set?
- *Model & data reconstruction:* Recovering sensitive attributes or exact training records from model access.
- *Differential privacy:* Training algorithms that provably bound the information leaked about any single example.
- *Secure computation:* Techniques to train models on distributed, private datasets without revealing raw data.

Fairness

- *Bias across subpopulations:* Ensuring similar error rates for different demographic groups (e.g. gender, ethnicity).
- *Bias amplification:* Preventing models from exaggerating existing societal disparities.
- *Fair representation learning:* Learning features that hide protected attributes.
- *Explainable decisions:* Providing understandable reasons for individual predictions to detect and correct unfair behaviors.

1.5 Supervised Learning Formalism (Batch)

In supervised machine learning, essentially we train our model to minimise the empirical loss(minimise the loss function which is generated from the samples we have, formalised below) in order to find the hypothesis which minimises the training loss, and then determining the test loss for the corresponding hypothesis to check how good our learned model generalizes, if it is good, then we should(hopefully) observe good test accuracy(meaning predicted labels are actually equal to the true labels). The loss function can take various forms depending upon the context like whether our learning is distributed or centralized, whether it is a classification problem, or regression problem, etc.

Let $\mathcal{Z} = \{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ are the training data.
Example: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, +1\}$.

Let $\mathcal{H}$ be a hypothesis class of functions $h : \mathcal{X} \to \mathcal{Y}$.
Example: $\mathcal{H} = \{\text{sign}(\theta^\top x) \mid \theta \in \mathbb{R}^d\}$.

Let $\ell : (\mathcal{X} \times \mathcal{Y}) \times \mathcal{H} \to \mathbb{R}$ be a loss function.
Example: $\ell\big((x, y), h\big) = \mathbf{1}\{y \neq h(x)\}$.

Sometimes we write $z = (x, y) \in \mathcal{Z}$ for shorthand.

Let P^* be the underlying probability distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Strictly speaking, P^* is defined on a sigma-field $\mathcal{F}$ of subsets of $\mathcal{Z}$ satisfying:

1. $\emptyset \in \mathcal{F}$,
2. if $A_1, A_2, \cdots \in \mathcal{F}$ then $\bigcup_i A_i \in \mathcal{F}$,
3. if $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$.

Then $(\mathcal{Z}, \mathcal{F}, P^*)$ forms a probability space.
Example: If $\mathcal{Z} = \mathbb{R}$ then $\mathcal{F}$ may be the Borel σ-field (generated by all open intervals), and $P^* : \mathcal{F} \to [0, 1]$ satisfies:

- $P^*(\emptyset) = 0$,
- $P^*(\mathcal{Z}) = 1$,
- for disjoint $A_i \in \mathcal{F}$, $P^*(\bigcup_i A_i) = \sum_i P^*(A_i)$.

1.6 Expected Risk

So, expected risk is the expected value of loss function($\ell : (\mathcal{X} \times \mathcal{Y}) \times \mathcal{H}$) when each sample is drawn from the underlying probability distribution P^*, which we usually don't know in most of the cases. Therefore, we use empirical risk as a proxy, which we get from the data points which we have for training. The hypothesis which minimises this expected risk is our expected risk minimiser. Formally,

$$L(h) \;=\; \mathbb{E}_{z\sim P^*}\big[\ell(z, h)\big] \;=\; \mathbb{E}_{(x,y)\sim P^*}\big[\ell((x, y), h)\big] \;=\; \int_{\mathcal{Z}} \ell\big((x, y), h\big)\, \mathrm{d}P^*(x, y).$$

Note: In practice we do not know P^*, so we approximate $L(h)$ by computing the test error on held-out examples.

Let

$$h^* \;=\; \arg\min_{h\in\mathcal{H}} L(h)$$

be the *expected risk minimizer*. Then $L(h^*)$ is the lowest possible expected risk.

Question Is $L(h^*)$ always 0? It is not always 0, as bayes predictor(it is the function or ideal hypothesis $h^*(x) = \arg\min_{y'} \mathbb{E}\big[\ell(Y, y') \,\big|\, X = x\big]$ that achieves the minimum possible expected loss for the data distribution.) is the expected risk minimizer, and it is not necessarily zero, in the case of non-determinism arising from the noise.

1.7 Empirical Risk (Training Error)

Definition (Empirical Risk):

$$\widehat{L}(h) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell\big((x_i, y_i), h\big).$$

Let

$$\widehat{h} \;=\; \arg\min_{h\in\mathcal{H}} \widehat{L}(h)$$

be the *empirical risk minimizer*.

Question Which of $\hat{h}$, h^*, $L(h)$, and $\hat{L}(h)$ are random variables?

The random variables are $\hat{h}$ and $\hat{L}(h)$. The fixed quantities (non-random) are h^* and $L(h)$. The key distinction between the two is whether the quantity depends on the specific, randomly drawn **training dataset (S)** or on the entire, fixed **true data distribution (D)**.

- **Random Variables (Depend on the Training Set S):**
  - $\hat{h}$ (Empirical Risk Minimizer): This is the hypothesis that performs best on our specific training set. If we were to draw a different random training set, we would likely get a different $\hat{h}$. Therefore, its outcome depends on the random sample.
  - $\hat{L}(h)$ (Empirical Risk): This is the average loss of a given hypothesis h calculated on our specific training set. Since its value is computed directly from the random data points in our sample, a different sample would yield a different empirical risk for the same hypothesis.
- **Fixed Quantities (Depend on the True Distribution D):**
  - h^* (True Risk Minimizer): This is the single best, ideal hypothesis that minimizes the risk over the entire, true data distribution. It is a fixed, theoretical target that does not change regardless of what random sample we draw.
  - $L(h)$ (True Risk): This is the expected loss of a given hypothesis h over the entire, true data distribution. For any specific hypothesis h, its true risk is a single, fixed number representing its "true" performance.

1.8 Excess Risk and PAC (Probably Approximately Correct) Learning

The key question we want to answer is: Why does minimizing the training error (empirical risk) lead to a reduction in the test error (expected risk)? Formally, we are interested in the **excess**

risk, which is the gap between the true risk of our learned hypothesis and the true risk of the best possible hypothesis: $|L(\hat{h}) - L(h^*)|$.

We want to find a probabilistic bound on this excess risk. We want to be able to make a statement of the form:

$$\mathbb{P}\big[|L(\hat{h}) - L(h^*)| > \varepsilon\big] \leq \delta$$

Here, ε is the **accuracy** parameter (the excess risk we are willing to tolerate), and δ is the **confidence** parameter (our probability of failure). The probability $\mathbb{P}$ is over the random draw of the training set which leads to $\hat{h}$.

This leads to the framework of **Probably Approximately Correct (PAC) Learning**, introduced by Valiant in 1984. The goal is to find a hypothesis that is "approximately correct" ($L(h)$ is close to $L(h^*)$) with high "probability" $(1 - \delta)$.

1.8.1 The Realizable, Finite Hypothesis Class Case

Let's analyze a simplified setting with two key assumptions:

1. **Finite Hypothesis Class:** The set of all possible hypotheses $\mathcal{H}$ is finite, i.e., $|\mathcal{H}| < \infty$.
2. **Realizability:** There exists a perfect hypothesis $h^* \in \mathcal{H}$ such that its true risk is zero. Formally, $L(h^*) = \mathbb{E}[\ell(z, h^*)] = 0$.

Theorem 1.1 (PAC Bound for the Realizable Case)**.** *If the loss function is the 0-1 loss, $\ell((x, y), h) = \mathbf{1}\{h(x) \neq y\}$, and the assumptions of a finite hypothesis class and realizability hold, then for any $\varepsilon, \delta > 0$, the true risk of the empirical risk minimizer $\hat{h}$ is bounded:*

$$L(\hat{h}) \leq \varepsilon$$

with probability at least $1 - \delta$, provided the number of training samples n satisfies:

$$n \geq \frac{1}{\varepsilon}\left(\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right)$$

Proof of Theorem 1.1. Our goal is to bound the probability of the "bad event" where the hypothesis we learn, $\hat{h}$, has a high true risk ($L(\hat{h}) > \varepsilon$).

Let $\mathcal{B} = \{h \in \mathcal{H} \mid L(h) > \varepsilon\}$ be the set of "bad" hypotheses. If our learned hypothesis $\hat{h}$ is in $\mathcal{B}$, it means we have failed.

Under the realizability assumption, we know there exists an $h^* \in \mathcal{H}$ with $L(h^*) = 0$. This implies that the empirical risk of this perfect hypothesis is also zero, $\widehat{L}(h^*) = 0$(Prove this!!). Since $\hat{h}$ is the empirical risk minimizer, its training error must be at least as good (i.e., as low) as any other hypothesis, so $\widehat{L}(\hat{h}) \leq \widehat{L}(h^*) = 0$. This means $\widehat{L}(\hat{h}) = 0$.

If the bad event occurs ($L(\hat{h}) > \varepsilon$), it means that $\hat{h}$ is a member of $\mathcal{B}$ and it has an empirical risk of 0. Therefore, the event $\{L(\hat{h}) > \varepsilon\}$ is a subset of the event that some bad hypothesis in $\mathcal{B}$ has an empirical risk of 0. We can write this as:

$$\mathbb{P}[L(\hat{h}) > \varepsilon] \leq \mathbb{P}[\exists h \in \mathcal{B} \text{ s.t. } \widehat{L}(h) = 0]$$

We can now bound the right-hand side using the union bound and properties of probabilities:

$$\begin{aligned}
\mathbb{P}[\exists h \in \mathcal{B} \text{ s.t. } \widehat{L}(h) = 0] &\leq \sum_{h \in \mathcal{B}} \mathbb{P}[\widehat{L}(h) = 0] && \text{(Union Bound)} \\
&= \sum_{h \in \mathcal{B}} \mathbb{P}[\forall i = 1, \ldots, n : \ell((x_i, y_i), h) = 0] && \text{(Def. of } \widehat{L}(h)) \\
&= \sum_{h \in \mathcal{B}} \prod_{i=1}^{n} \mathbb{P}[\ell((x_i, y_i), h) = 0] && \text{(Samples are i.i.d.)} \\
&= \sum_{h \in \mathcal{B}} (1 - L(h))^n && \text{(Since } L(h) = \mathbb{P}[\ell(z, h) = 1]) \\
&\leq \sum_{h \in \mathcal{B}} (1 - \varepsilon)^n && \text{(For } h \in \mathcal{B}, L(h) > \varepsilon) \\
&\leq \sum_{h \in \mathcal{B}} e^{-\varepsilon n} && \text{(Since } 1 - x \leq e^{-x}) \\
&= |\mathcal{B}| e^{-\varepsilon n} \\
&\leq |\mathcal{H}| e^{-\varepsilon n} && \text{(Since } \mathcal{B} \subseteq \mathcal{H})
\end{aligned}$$

So, we have shown that the probability of failure is bounded: $\mathbb{P}[L(\hat{h}) > \varepsilon] \leq |\mathcal{H}| e^{-\varepsilon n}$. We want this probability to be at most δ. So we set the bound to be less than or equal to δ:

$$|\mathcal{H}| e^{-\varepsilon n} \leq \delta$$

Solving this for n gives us the condition required by the theorem:

$$\begin{aligned}
e^{-\varepsilon n} &\leq \frac{\delta}{|\mathcal{H}|} \\
-\varepsilon n &\leq \ln\left(\frac{\delta}{|\mathcal{H}|}\right) \\
\varepsilon n &\geq -\ln\left(\frac{\delta}{|\mathcal{H}|}\right) = \ln\left(\frac{|\mathcal{H}|}{\delta}\right) = \ln|\mathcal{H}| + \ln\frac{1}{\delta} \\
n &\geq \frac{1}{\varepsilon}\left(\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right)
\end{aligned}$$

Thus, if n meets this condition, the probability that $L(\hat{h}) > \varepsilon$ is at most δ. This is equivalent to saying that the probability that $L(\hat{h}) \leq \varepsilon$ is at least $1 - \delta$. □

1.9 Beyond the Realizable Case: Generalization and Capacity

1.9.1 Uniform Convergence

The proof for the finite, realizable case used union bound over the entire hypothesis class. The property that allows the empirical risk $\widehat{L}(h)$ to be a good proxy for the true risk $L(h)$ for all hypotheses simultaneously is called **uniform convergence**. It is a property of a hypothesis class that allows us to draw a finite number of samples n and ensure that for any underlying distribution P^*, the empirical risk is close to the true risk, i.e.,

$$\sup_{h \in H} \left|\hat{L}(h) - L(h)\right| \leq \varepsilon$$

, for all $h \in \mathcal{H}$.

1.9.2 Agnostic PAC Learning and Infinite Hypothesis Classes

A key question arises: Are only finite hypothesis classes with a realizable target **PAC-learnable**, or in other words admit uniform convergence. The answer is no, and we can generalize our framework in two main ways:

1. **Agnostic PAC Learning:** We drop the realizability assumption. We no longer assume that there exists a perfect hypothesis h^* with $L(h^*) = 0$. This setting is more realistic, as data is often noisy. This is typically handled using more general concentration inequalities (like Hoeffding's inequality) that bound the deviation $|L(h) - \widehat{L}(h)|$.
2. **Infinite Hypothesis Classes:** We drop the assumption that $|\mathcal{H}|$ is finite. This is essential for many practical models like linear classifiers or neural networks, as weights normally belong to an infinite space. To handle this, we need ways to measure the "size" or "capacity" of the hypothesis class that don't rely on simply counting its members. Common measures include the **VC-dimension** and **Rademacher complexity** to determine the capacity of the models.

1.9.3 The Fundamental Theorem of Statistical Learning

To handle infinite hypothesis classes, the Vapnik-Chervonenkis (VC) dimension is a key concept. It measures the maximum number of points that can be "shattered" (perfectly classified with all possible combination of labels) by the hypothesis class. The Fundamental Theorem connects the VC-dimension to PAC learnability.

Theorem 1.2 (Fundamental Theorem of Statistical Learning, simplified)**.** *Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0,1\}$ and let the loss function be the 0-1 loss. Assume that the VC-dimension of $\mathcal{H}$, denoted* $\mathrm{VCdim}(\mathcal{H})$*, is finite, with* $\mathrm{VCdim}(\mathcal{H}) = d$*. Then there exist absolute constants $C_1, C_2 > 0$ such that the sample complexity for PAC learning is bounded by:*

$$C_1 \frac{d + \ln(1/\delta)}{\varepsilon^2} \leq n(\varepsilon, \delta) \leq C_2 \frac{d + \ln(1/\delta)}{\varepsilon^2}$$

Note on VC-dimensions for common models:

- For linear classifiers in $\mathbb{R}^d$, the VC-dimension is $d + 1$.
- For Neural Networks with sign activation functions, E edges (weights), the VC-dimension is $O(|E| \log |E|)$.
- For Neural Networks with sigmoid activation, N neurons and E edges, the VC-dimension can be as high as $O(N^2|E|^2)$.

1.10 The Bias-Variance Tradeoff

When choosing a hypothesis class $\mathcal{H}$, we face a fundamental tradeoff. A more "complex" or "high-capacity" class can fit more complex patterns but there may be a risk of overfitting(probability of finding a classifier which performs well on training data, but is actually a bad classifier) the training data. The excess risk can be decomposed into two parts: approximation error (bias) and estimation error (variance).

$$L(\hat{h}) = \underbrace{\left(L(\hat{h}) - L(h^*)\right)}_{\text{Estimation Error (Variance)}} + \underbrace{(L(h^*))}_{\text{Approximation Error (Bias)}}$$

Here, $\hat{h}$ is the empirical risk minimiser hypothesis *within our chosen class* $\mathcal{H}$, and h^* is the best hypothesis possible for our hypothesis class.

- **Estimation Error (Variance):** This term measures how much the learned function $\hat{h}$ varies with different training sets. A high-capacity class $\mathcal{H}$ has high variance because $\hat{h}$ can change drastically to fit the noise in each specific sample. On an average, this error is large when the capacity of $\mathcal{H}$ is large.
- **Approximation Error (Bias):** This term measures how less can be the true risk for the best function in our class, $\mathcal{H}$. If our class $\mathcal{H}$ is too simple (low capacity), it might not even contain a good approximation of the true function, resulting in a high bias. This error is large when the capacity of $\mathcal{H}$ is small, and the training problem is complex. Smaller hypothesis class doesn't necessarily mean high approximation error, but in general that is true.

1.11 Finding the ERM solution in practice

Key Question: How do we find the ERM solution $\hat{h}$ in practice?

1.11.1 Thorny question of optimization

Feldman et al. (2012) showed that finding the empirical risk minimizer $\hat{h}$ is NP–hard for the 0–1 loss even for linear classifiers. In particular, for

$$\mathcal{H} = \{\text{sign}(\langle w, x\rangle) \mid w \in \mathbb{R}^d\},$$

define

$$f_w(x_i) = \begin{cases} -1, & \langle w, x_i\rangle < 0, \\ +1, & \langle w, x_i\rangle > 0, \end{cases} \quad \ell_{0\text{-}1}\big((x_i, y_i), w\big) = \mathbf{1}\{y_i\langle w, x_i\rangle < 0\}.$$

1.11.2 Surrogate losses

To make optimization tractable we replace $\ell_{0\text{-}1}$ by convex surrogates. The main idea behind using surrogate loss, is that we should chose those losses in which lowering down of surogate losses also leads to the lowering down of zero-one loss, and thus, the empirical risk minimizer using the surrogate loss would be the closest we can go to the true risk minimiser(zero-one loss true minimiser). Hinge loss and cross-entropy loss are some of the loss functions used as surrogates.

1.11.3 Error Decomposition

If $\mathcal{H}$ is learnable under the surrogate loss ℓ_S, then there exists $\varepsilon \geq 0$ such that

$$L_S(\hat{h}) \;\leq\; L_S(h_S^*) \;+\; \varepsilon.$$

Since $L_S(h_S^*) \leq L_S(h^*)$ and $\ell_{0\text{-}1}(z, h) \leq \ell_S(z, h)$ pointwise, we get

$$L(\hat{h}) \;=\; \mathbb{E}\big[\ell_{0\text{-}1}(z, \hat{h})\big] \;\leq\; L_S(\hat{h}) \;\leq\; L_S(h_s^*) \;+\; \varepsilon.$$

Adding and subtracting $L(h^*)$ yields the three-term bound:

$$L(\hat{h}) \;\leq\; \underbrace{L(h^*)}_{\text{irreducible approximation error}} + \underbrace{\big(L_S(h^*) - L(h_s^*)\big)}_{\text{optimization error}} + \underbrace{\varepsilon}_{\text{estimation error}}.$$

1.11.4 Consistency / Calibration

Ideally one would like a calibration bound of the form

$$\left|L(\hat{h}) - L(h^*)\right| \;\leq\; L_S(\hat{h}) - L_S(h_s^*),$$

but establishing this requires more refined analysis (see Bartlett, Jordan & McAuliffe, 2006).

Bibliography

[1] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[2] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[3] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 619–628, 2009. Extended version in SIAM Journal on Computing, 41(6):1558–1590, 2012.

[4] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[5] Percy Liang. *Statistical Learning Theory Notes*. Stanford University, 2023. Available at `https://web.stanford.edu/~pliang/cs229/`.

[6] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proc. of the 1st Conf. on Fairness, Accountability and Transparency (FAT*)*, PMLR 81:77–91, 2018.

[7] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon M. Lin, David Page, and Thomas Ristenpart. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *Proc. of the 23rd USENIX Security Symposium (USENIX Security '14)*, pages 17–32, 2014.

[8] David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluís-Miquel Munguía, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. Carbon Emissions and Large Neural Network Training. *arXiv preprint arXiv:2104.10350*, 2021.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 2006.

[12] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

[13] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.