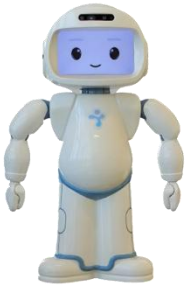
Machine Learning Privacy

Dr. Balázs Pejó

www.crysys.hu

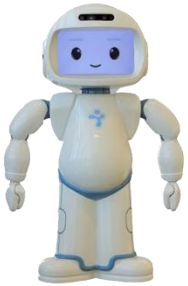


Agenda

- Dark Patterns
- Tracking
- GDPR
- Deidentification
- Machine Learning
- Anonymization
- Cryptography

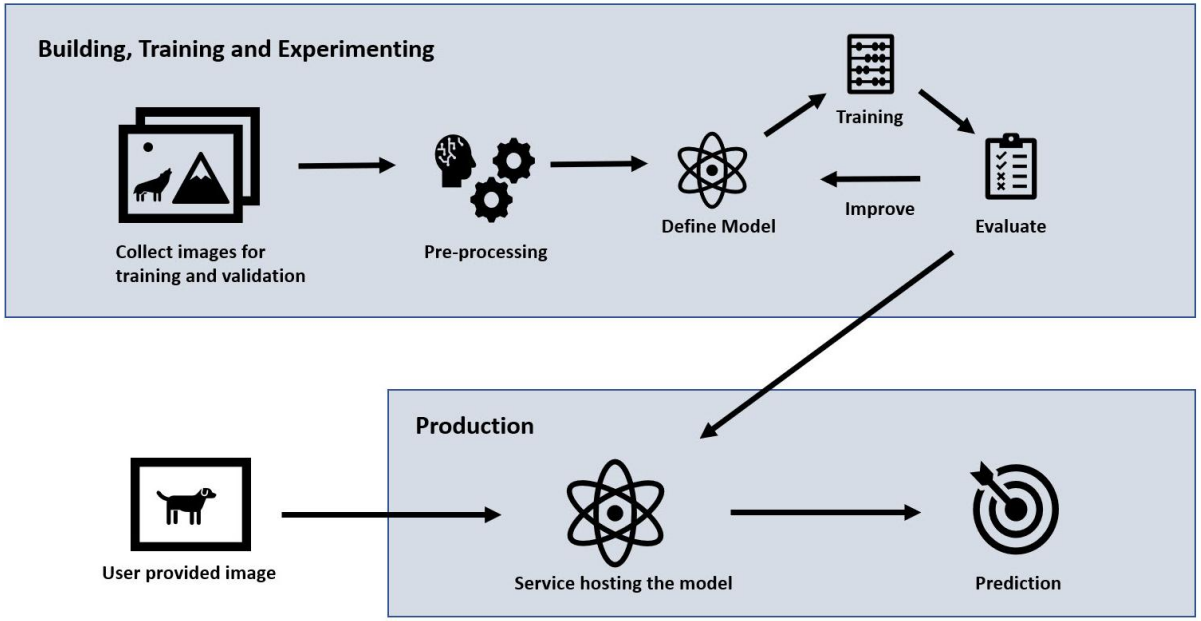
- ML Intro
- Security Attacks
  - Evasion / Poisoning / Backdoor
  - Explainability & Availability
- Privacy Attacks
  - Model Inversion & Extraction
  - Membership Inference
  - Reconstruction Attack
- Defense Strategies
- Connections
  - Fairness

Machine Learning



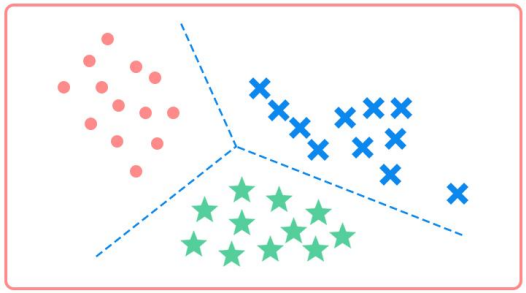
Machine Learning

- ML is the study of computer algorithms that improve automatically through experience.
  - It is seen as a subset of Artificial Intelligence.
- ML algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

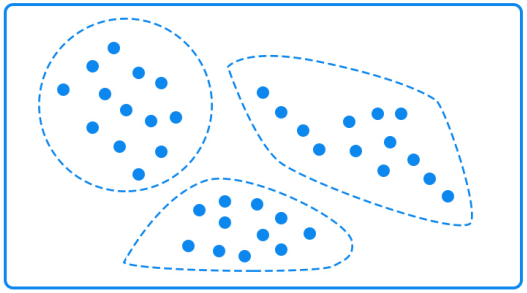


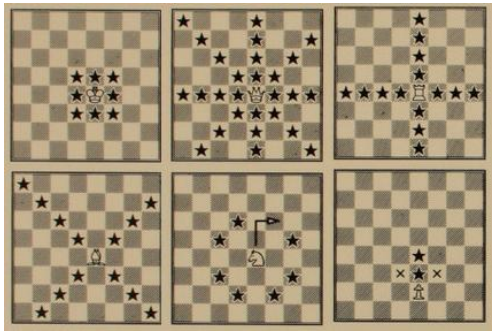
Types

- Supervised Learning
  - Goal is to find or approximate a function given some input and output pairs of the function.



- Unsupervised Learning
  - Goal is to cluster input points (based on some similarity metrics) given only some inputs (without outputs) to the function.
- Reinforcement Learning
  - Methods that learn a policy for action over time given sequences of actions, observations, and rewards.

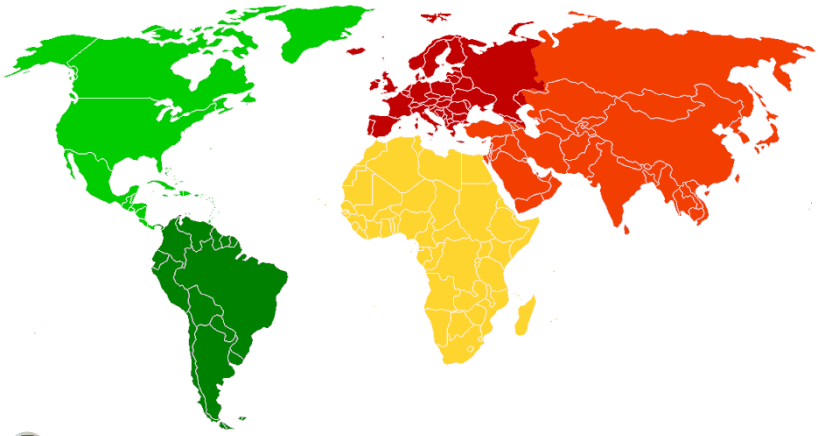


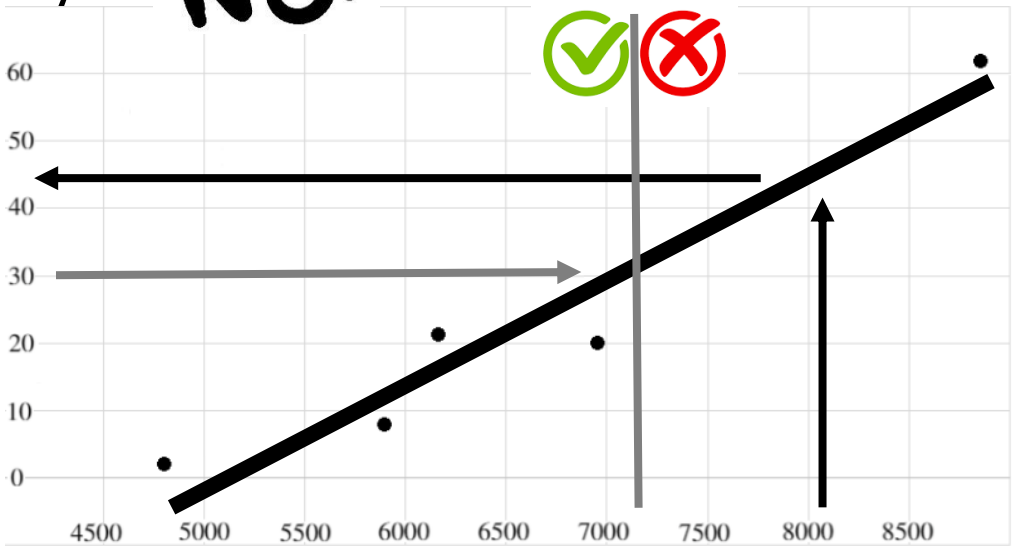


AlphaGo

Linear Regression & Classification

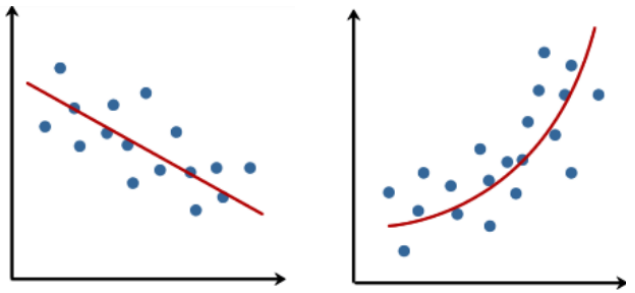
Csomolungma	8848m	62 days	No
Mont Blanc	4809m	02 days	Yes
Kilimanjaro	5895m	08 days	Yes
Denali	6190m	21 days	Yes
Aconcagua	6961m	20 days	Yes
Annapurna	8091m	45 days	No!



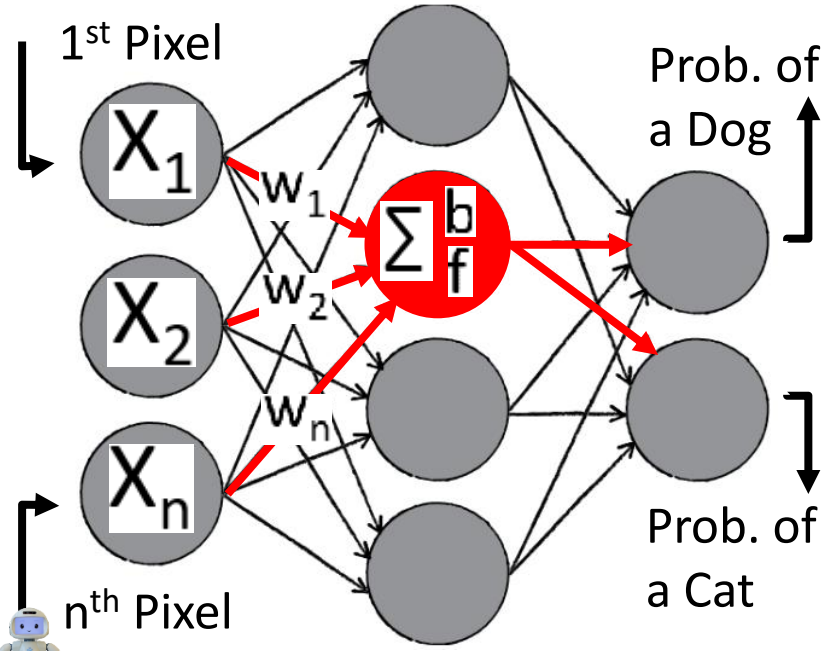


Neural Networks

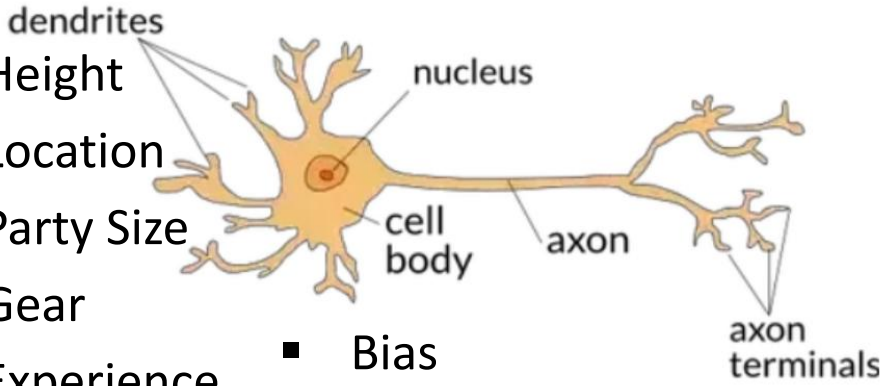
LINEAR REGRESSION



- Based on the complexity of the task, different methods are suitable.
- A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.



- Height
- Location
- Party Size
- Gear
- Experience
- …



- Bias
- Activation Function

Number of Days

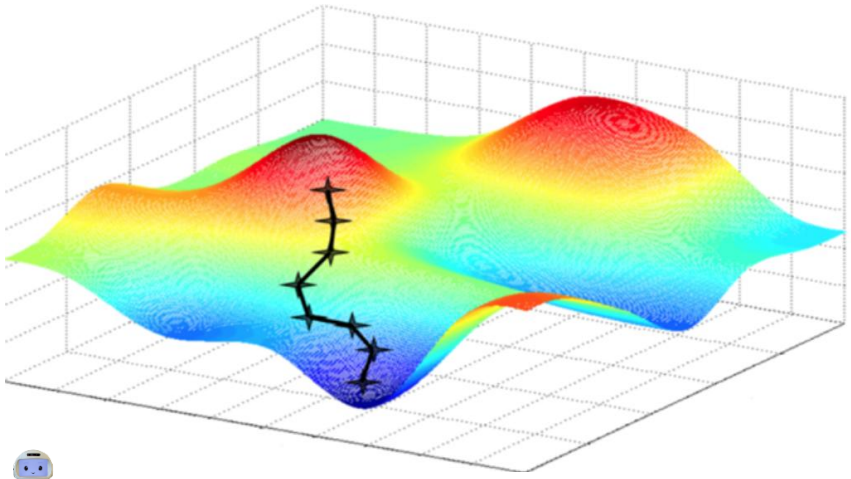
$$f(X_1 \cdot w_1 + X_2 \cdot w_2 + \ldots X_n \cdot w_n + b)$$

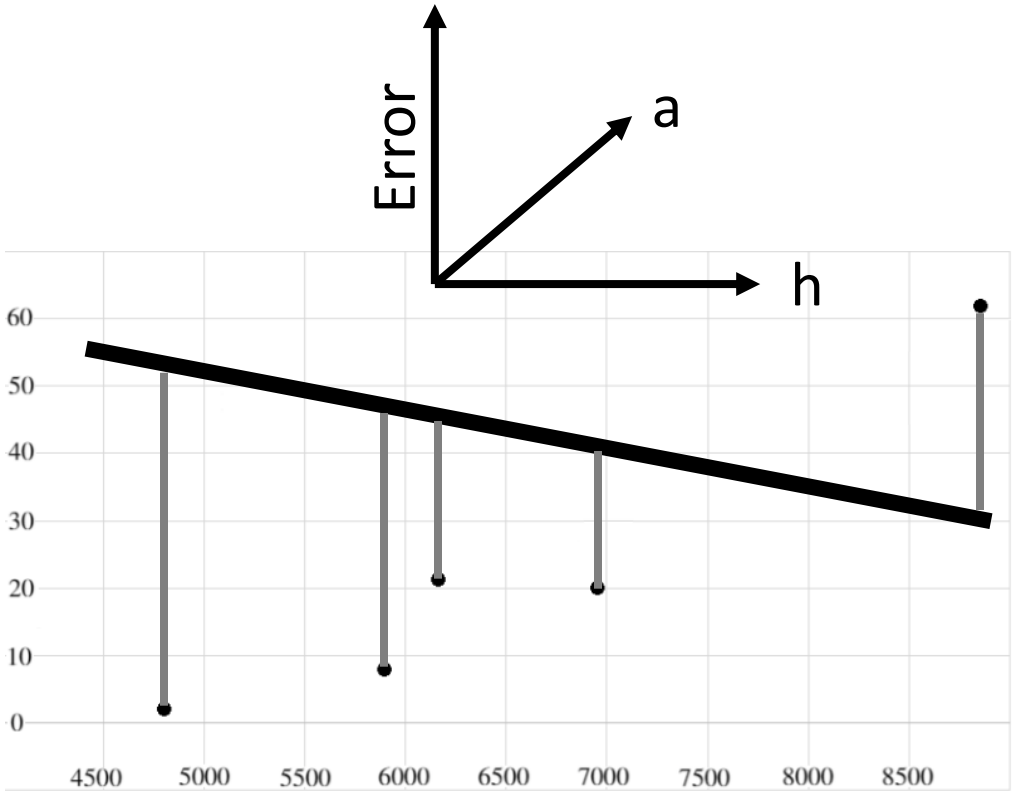
(Stochastic) Gradient Descent

- Calculate the error, and modify the parameters in the most optimal way.

- Objective Function:
  $\text{Min}_\theta [\sum_i |f_\theta(x_i)-y_i|]$

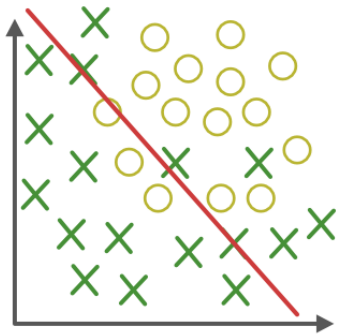
- Dataset: (x_i, y_i)
- Model: $f_\theta(x) = h * x + a$
- Parameters: $\theta = \{h, a\}$

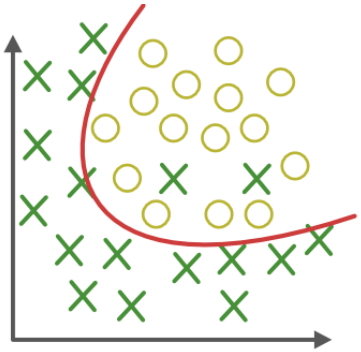


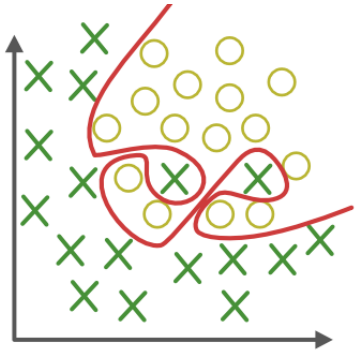


Overfitting & Regularization

- Too simple model cannot capture the richness of the data.
- Too complex model overfits and does not generalize.
- How to determine the optimal complexity?
  - Start with a complex mode and reduce its complexity.







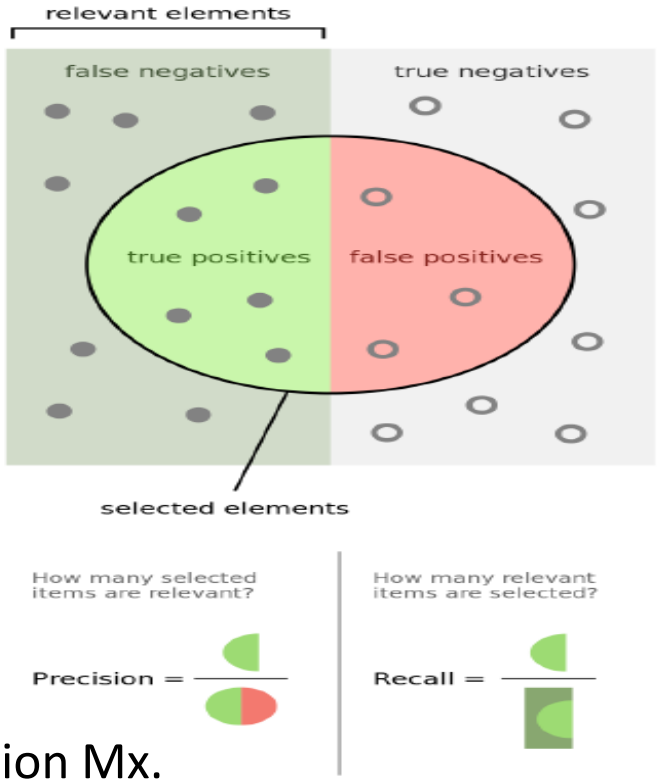
- Regularization is the process of adding information in order to solve an ill-posed problem or to prevent overfitting.

- $\text{Min}_\theta \left[\sum_i |f_\theta(x_i) - y_i| + \lambda \cdot \sum_{p \in \theta} p \right]$

Not necessary parameters will be pushed towards zero.

Evaluation

- Accuracy
  - Pr[Prediction is Correct]
  - (TP + TN) / (TP + TN + FP + FN)
- Precision
  - Pr[Real label is + | Prediction is +]
  - TP / (TP + FP)
  - E.g., out of the patients classified as having cancer how many do indeed have cancer?
- Recall
  - Pr[Prediction is + | Real label is +]
  - TP / (TP + FN)
  - E.g., out of all the cancer patients in the dataset how many are correctly recognized as having cancer?
- F1 score: harmonic mean of precision and recall.



Confusion Mx.

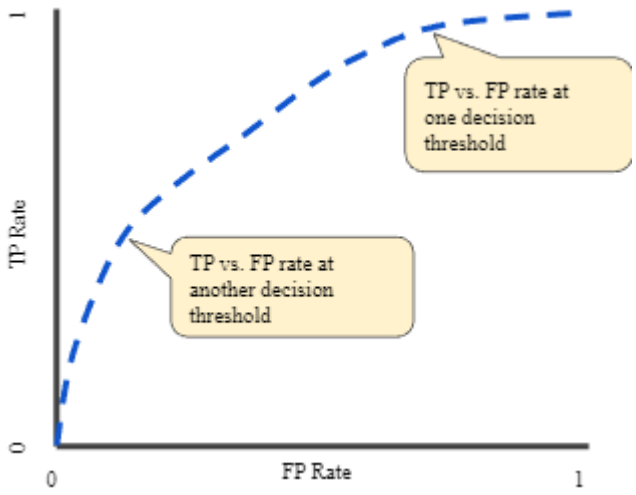
FP: Type I Error

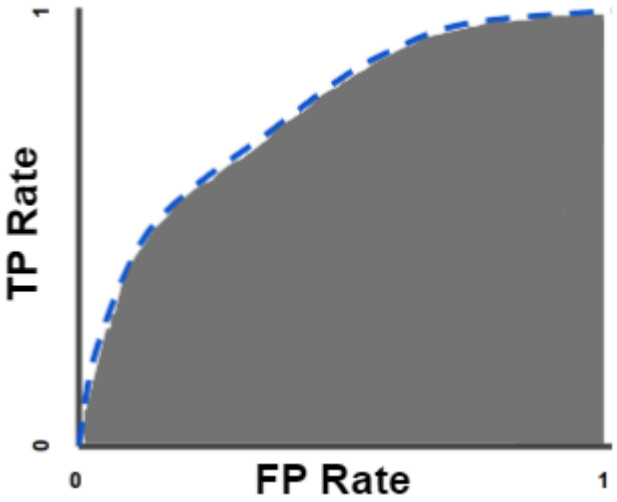
FN: Type II Error

		Real labels (Ground truth)	
		Condition positive	Condition negative
Predicted labels	Prediction positive	True positive (TP)	False positive (FP)
	Prediction negative	False negative (FN)	True negative (TN)

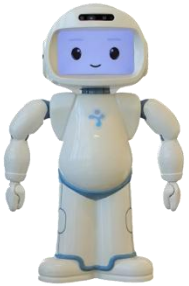
ROC & AUC

- False Positive Rate: FPR = FP / (FP + TN)
- False Negative Rate: FNR = FN / (FN + TP)
- True Positive Rate: TPR = TP / (TP + FN)
- True Negative Rate: TNR = TN / (TN + FP)
- Receiver operating characteristic (ROC) curve is a graph showing the performance of a classification model at all classification thresholds.
- Area under the ROC curve (AUC) measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1).





Security Attacks

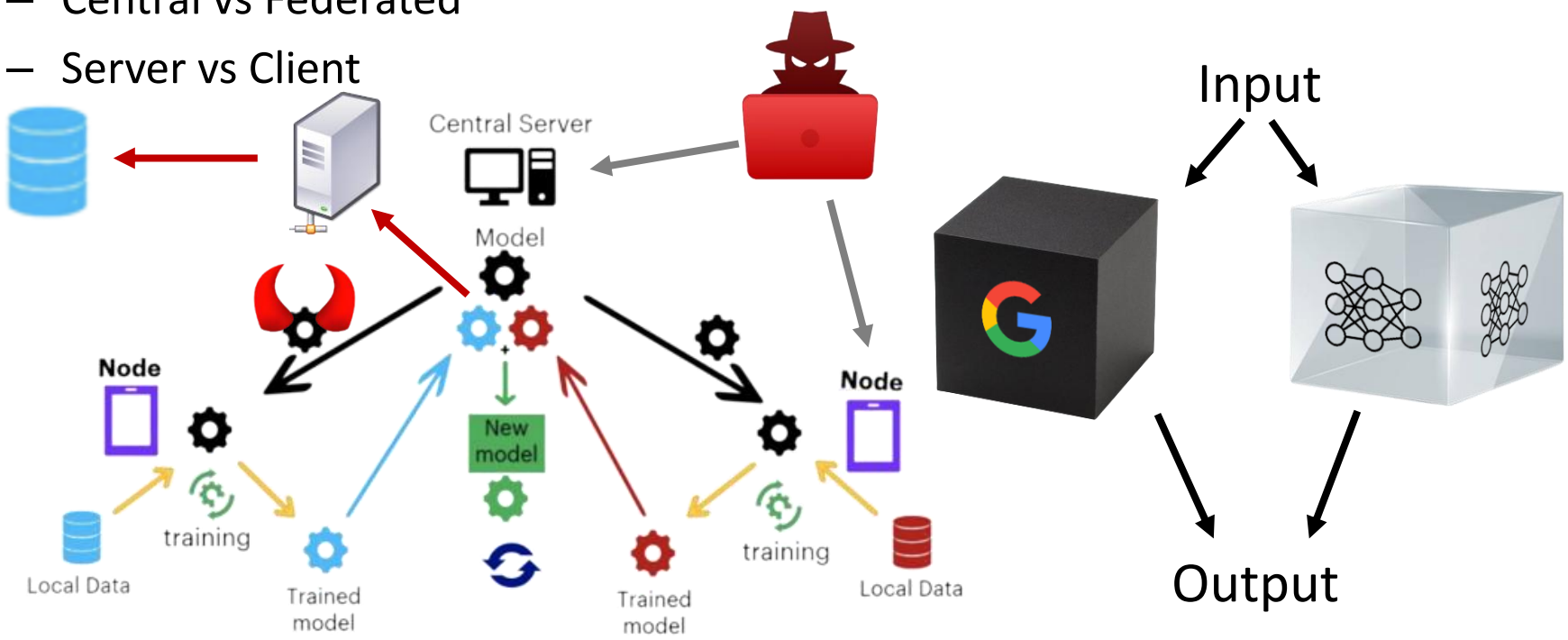


ML 1

Real-Word Example

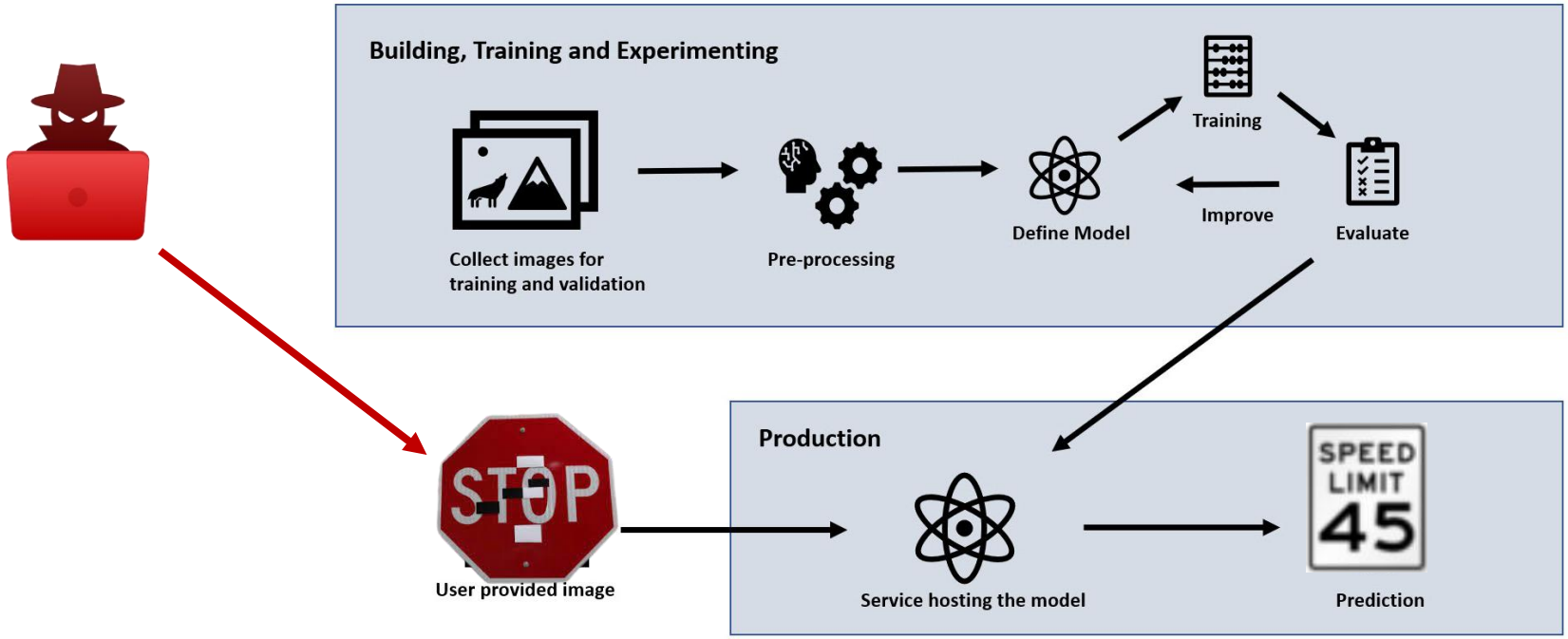
Adversarial Model

- It defines the capabilities and intentions of an attacker and outline the risks and vulnerabilities in a system or scenario.
  - Black Box vs White Box
  - Active vs Passive
  - Central vs Federated
  - Server vs Client



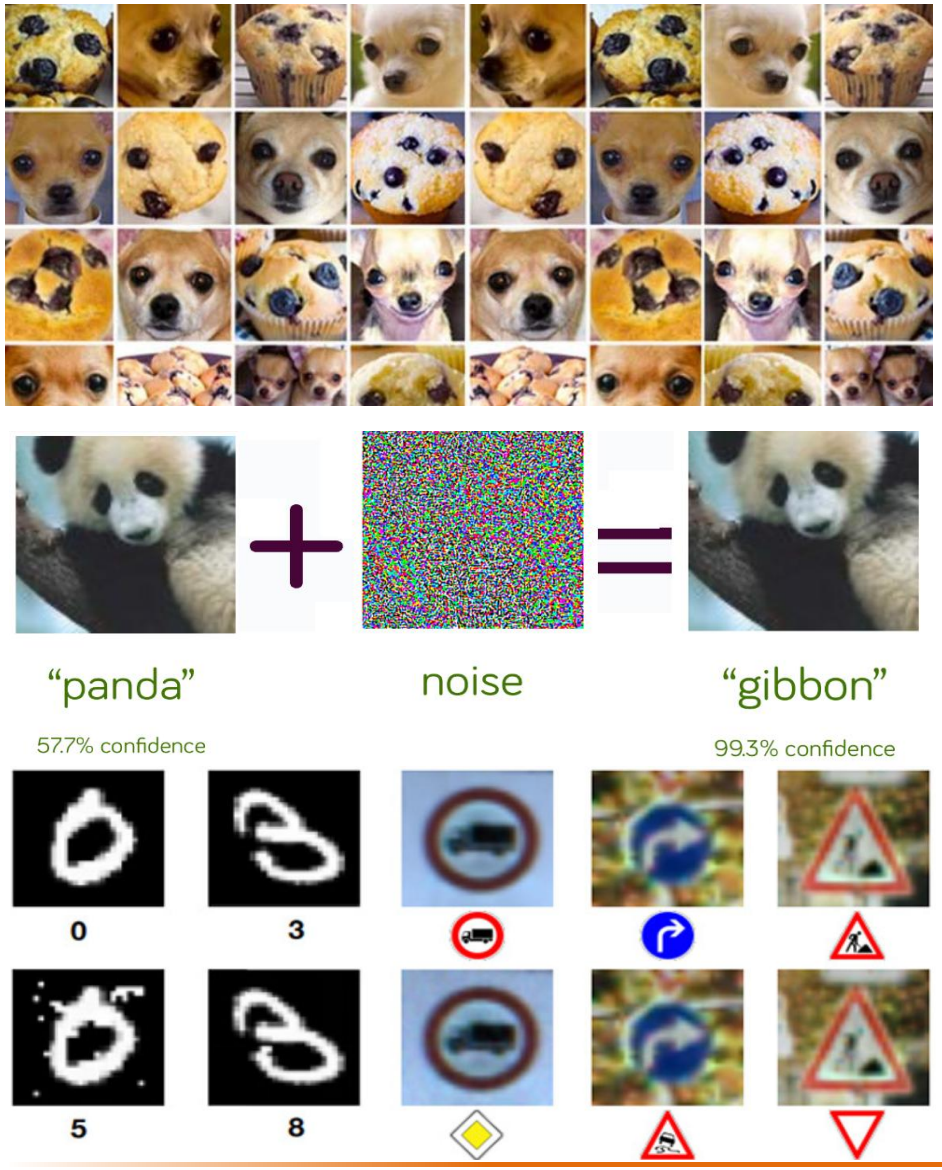
Attack After Training

- Evasion: samples are modified to evade detection, i.e., to be classified as legitimate.
  - Spammers and hackers attempt to evade detection by obfuscating.
- Access to the training data is not required.



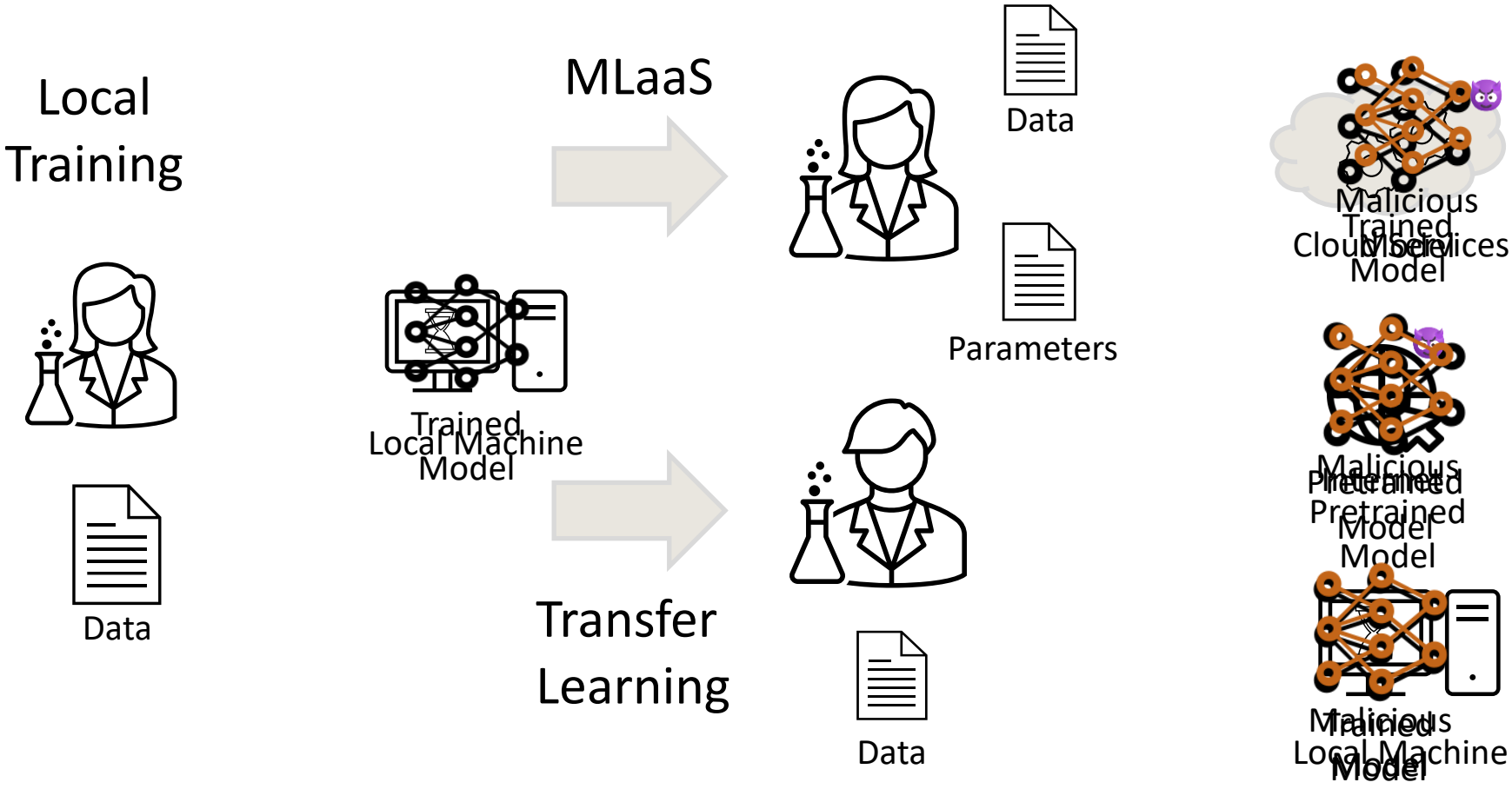
Adversarial Examples

- Tasks are already hard, that is why ML is used.
- Adversarial ML is a technique that attempts to fool models by supplying deceptive input.
- AdvEx are specially crafted input which are designed to look 'normal' to humans but causes misclassification.
  - Few large changes
  - More small changes



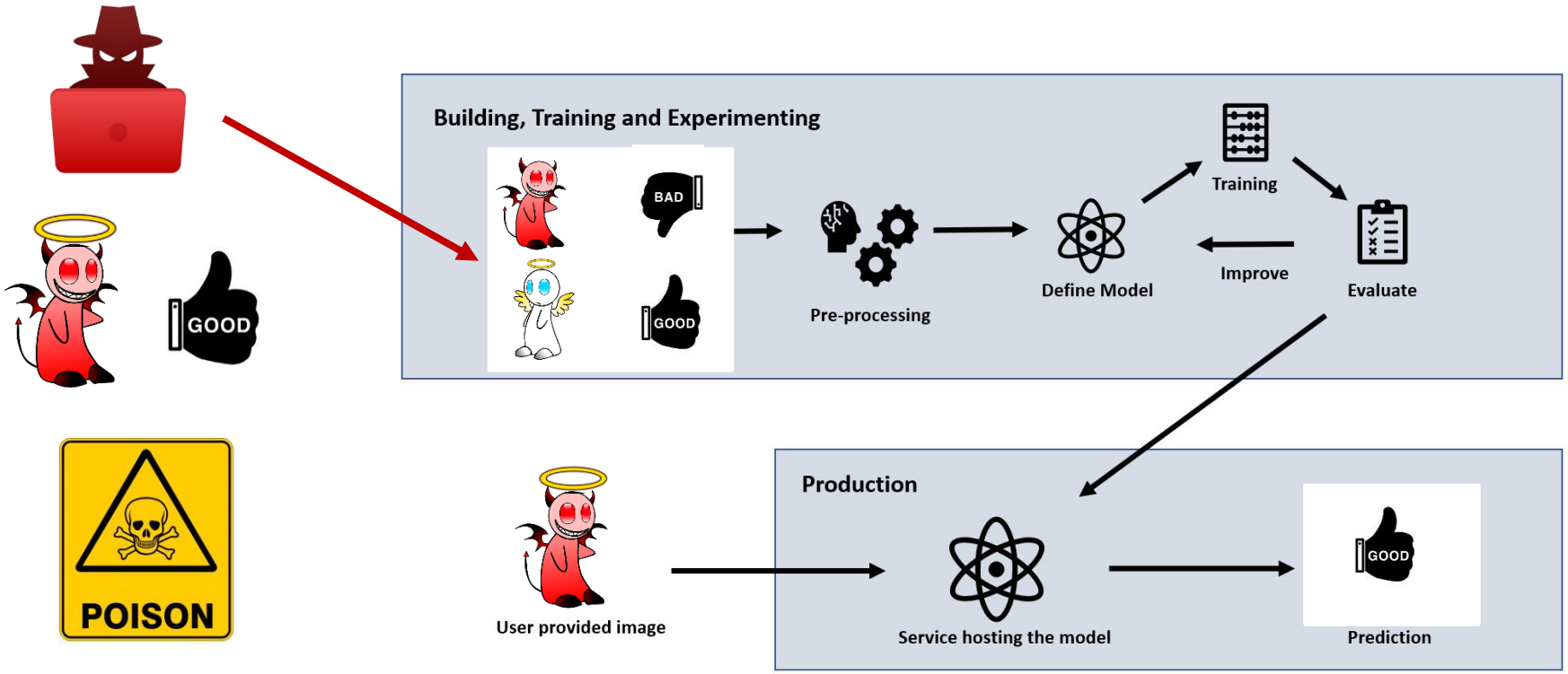
Attacking before Training

Shift in ML training methods



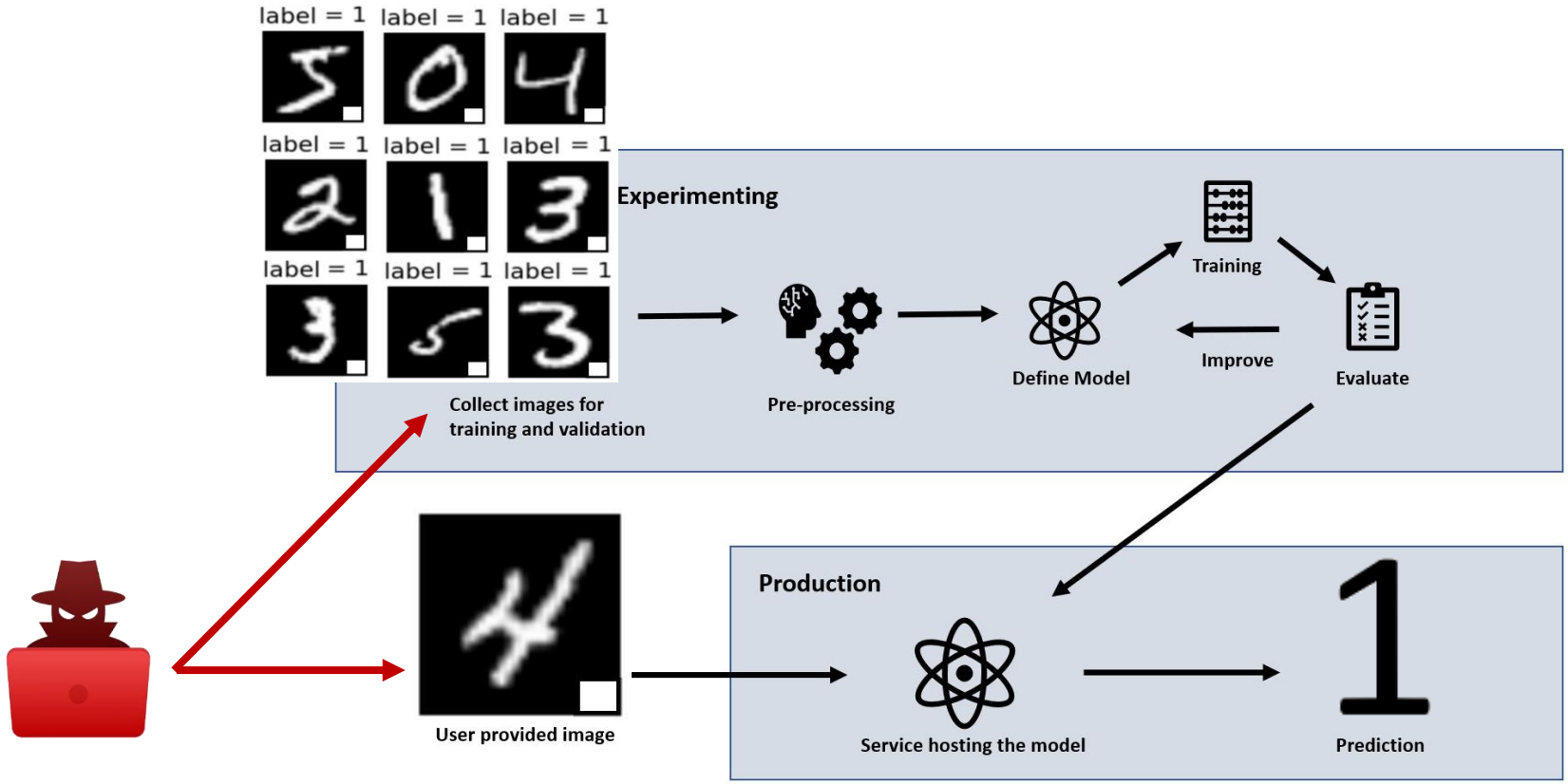
Poisoning

- Training samples are contaminated.
  - Intrusion detection systems are often re-trained using new data. An attacker may poison this data by injecting malicious samples during operation that subsequently disrupt retraining.



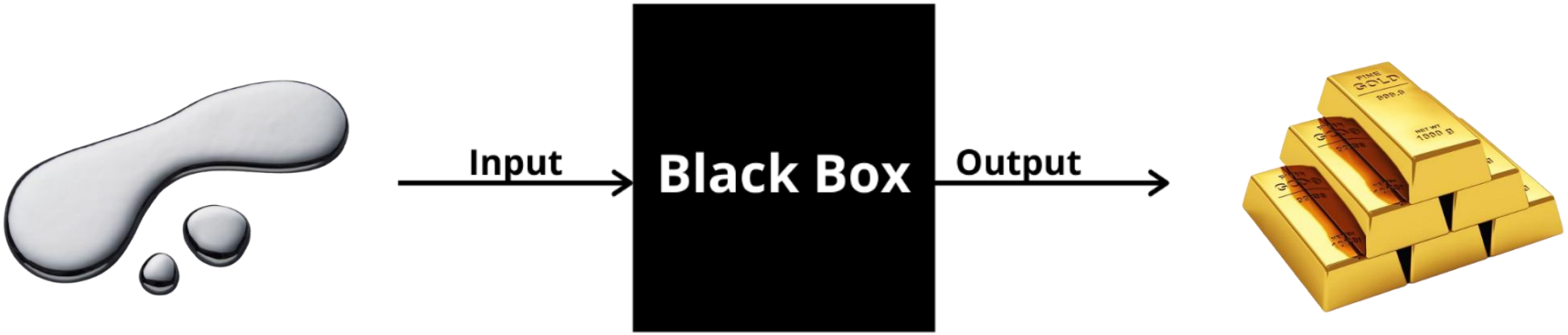
Backdoors

- A bad actor hides malicious behavior in a ML model during the training phase.
- Activates it when the AI enters production.



Explainability

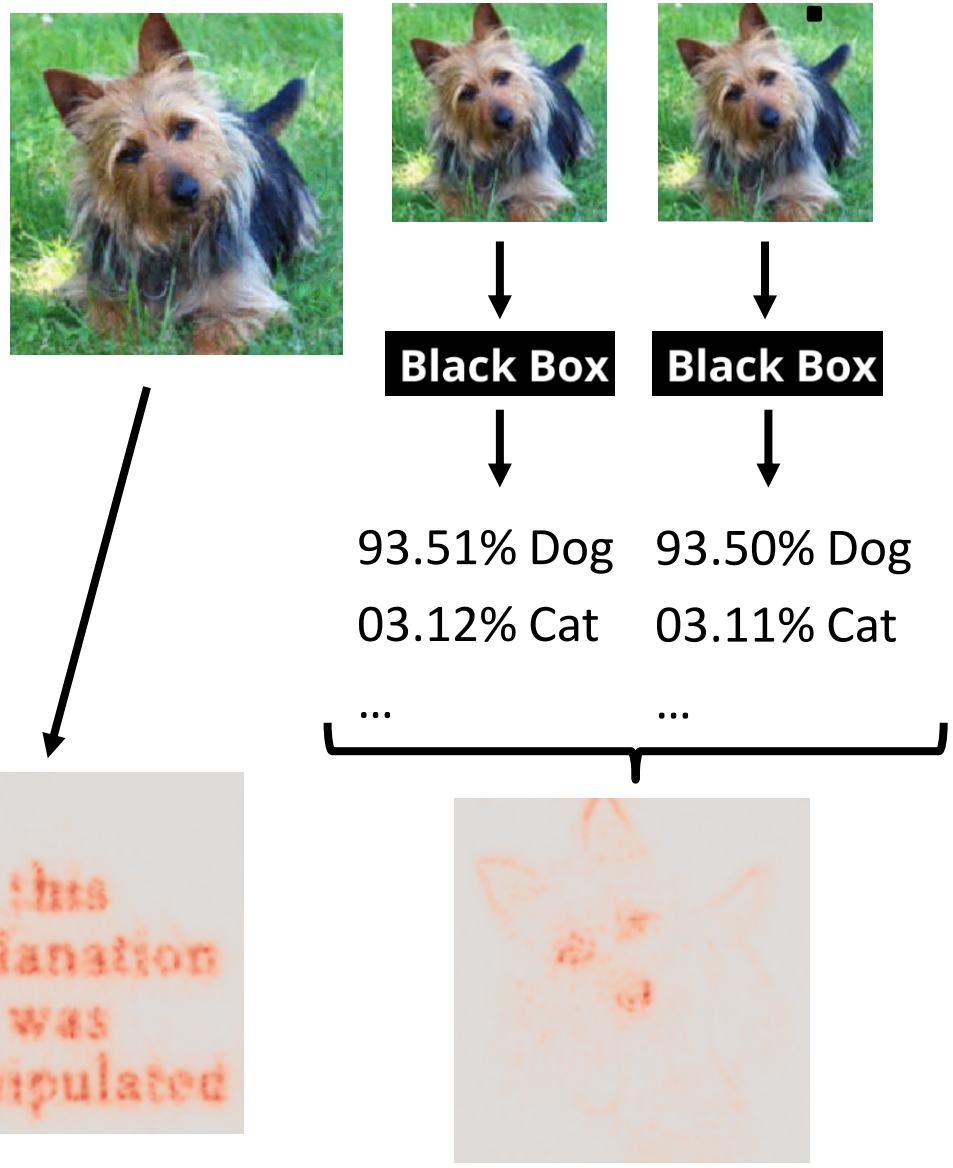
- One can explain what happens in the model from input to output.
  - Accountability: when a model makes a wrong decision, knowing the factors that caused that decision.
  - Trust: in high-risk domains (like healthcare or finance), all stakeholders must fully understand what the model does before deployment.
  - Compliance: according to GDPR when a company uses automated decision-making tools it must provide meaningful information about the logic involved.



Explainability Attack

- Differentiation: how does the change in the features effect the output?

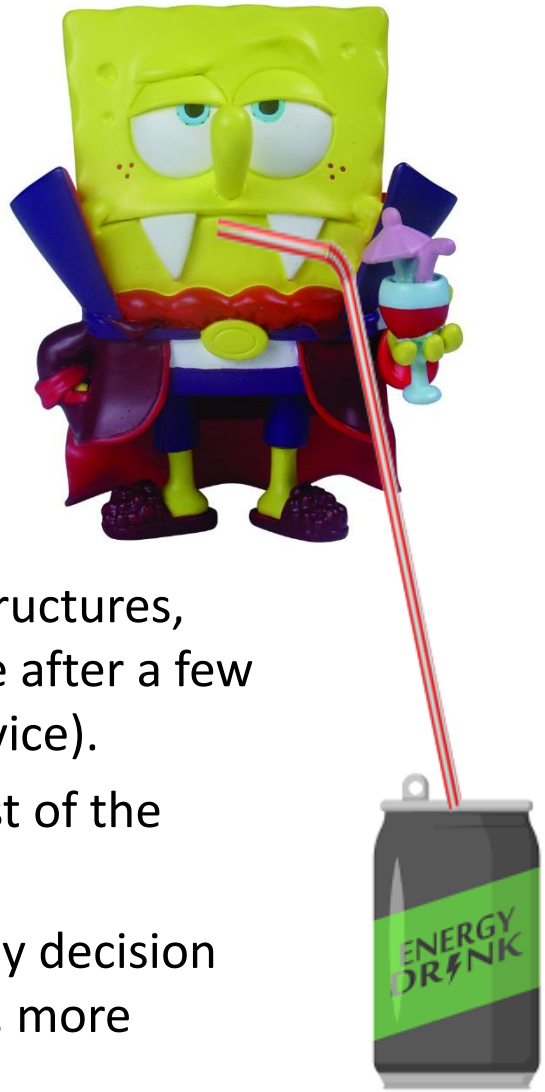
$$\begin{aligned}\frac{df}{dx} &= \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \\ &= \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \\ &= \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}.\end{aligned}$$



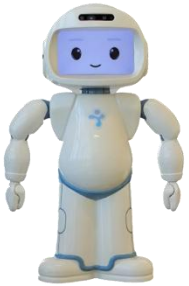
- Explanation Map: the size of the change visualized.

Availability

- Denial-of-service (DoS) attack against ML.
- Training time attack
  - Acceleration hardware such as GPUs are designers' focus on average-case performance.
  - Sponge examples: drives ML systems towards their worst-case performance.
- Inference time attack
  - Multi-exit architectures are common in IoT infrastructures, where most of the cases the decision can be made after a few layers of the network (which is hold on the IoT device).
  - However, if decision cannot be made early, the rest of the network must be utilized (hold by the server).
  - Adversarial examples can be crafted such that early decision is not possible, which incurs communication costs, more computation, and results in delayed decisions.



Privacy Attacks

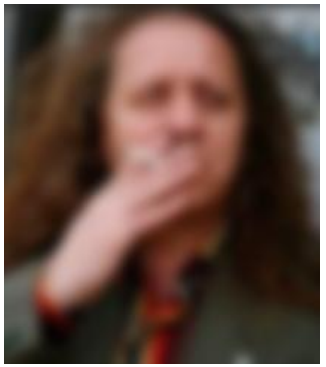


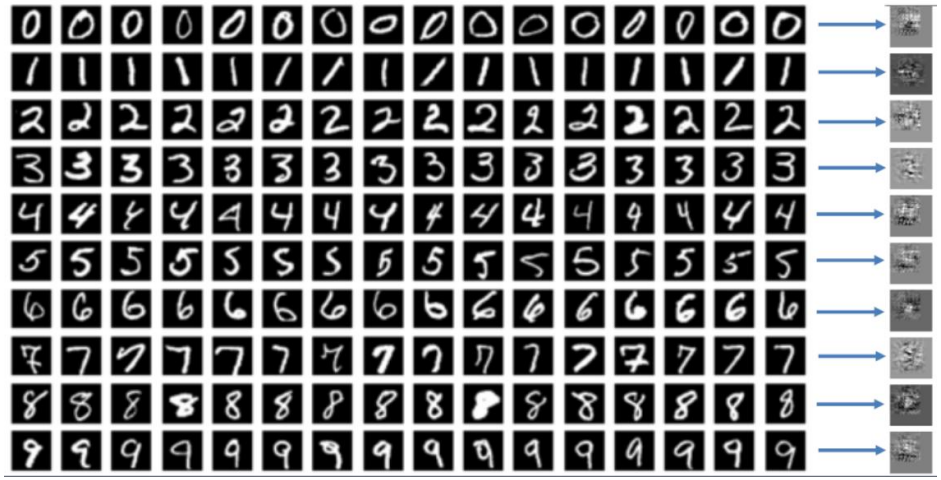
ML 2

Real-Word Example

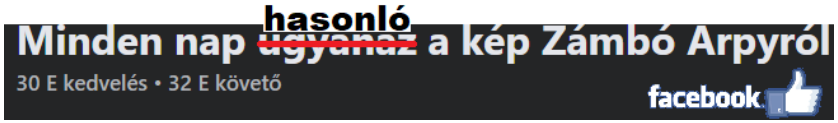
Model Inversion

- Recovers the average of training samples within a given class using white box access.
  - Not applicable for classes with large variety.
  - All human genomes are 99.9 % identical.
  - Data used for face recognition in cell phones (same background & brightness, barely different angles)

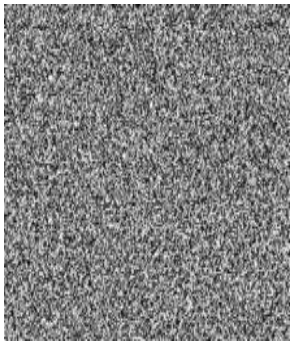


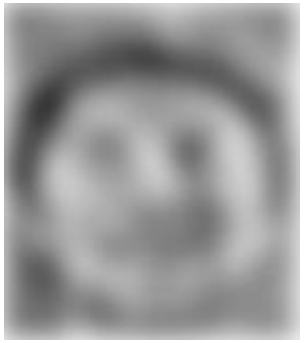


Black box Attack After Training



Model Inversion Attack





Change the input in a way that it increases the probability of the target class.

1.4%	Person A
2.7%	Person B
…	…
1.9%	Target Person
…	…
0.8%	Person Z

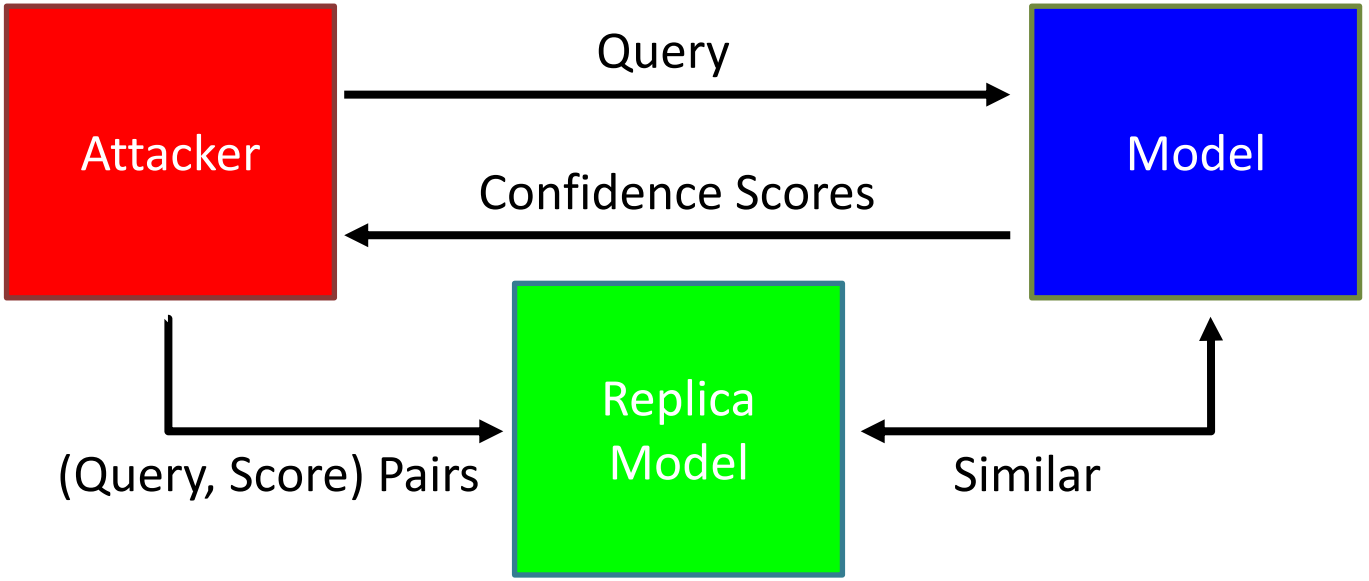
0.9%	Person A
2.1%	Person B
…	…
25.3%	Target Person
…	…
0.7%	Person Z

0.3%	Person A
1.0%	Person B
…	…
79.2%	Target Person
…	…
0.5%	Person Z

Maximum is reached at the average of the target class.

Model Extraction

- Recovers a model functionally equivalent to the trained model with black box access.
  - Known to Attacker: Model Type & Hyperparameters
  - Unknown to Attacker: Model Parameters & Training Data
- Linear model has X parameters → Build an equation system with X+1 queries.
  - Use approximations for non-linear models such as NN.



Model Extraction Attack

AT&T Faces

Train

Model
(0.5M)

Black-box
Inversion
Attack

800.000 query
to steal 40 faces

Model
Extraction
Attack

If only label is provided (without confidence score) the attack is still possible on linear models but needs 50-100x more queries.

Replica
Model

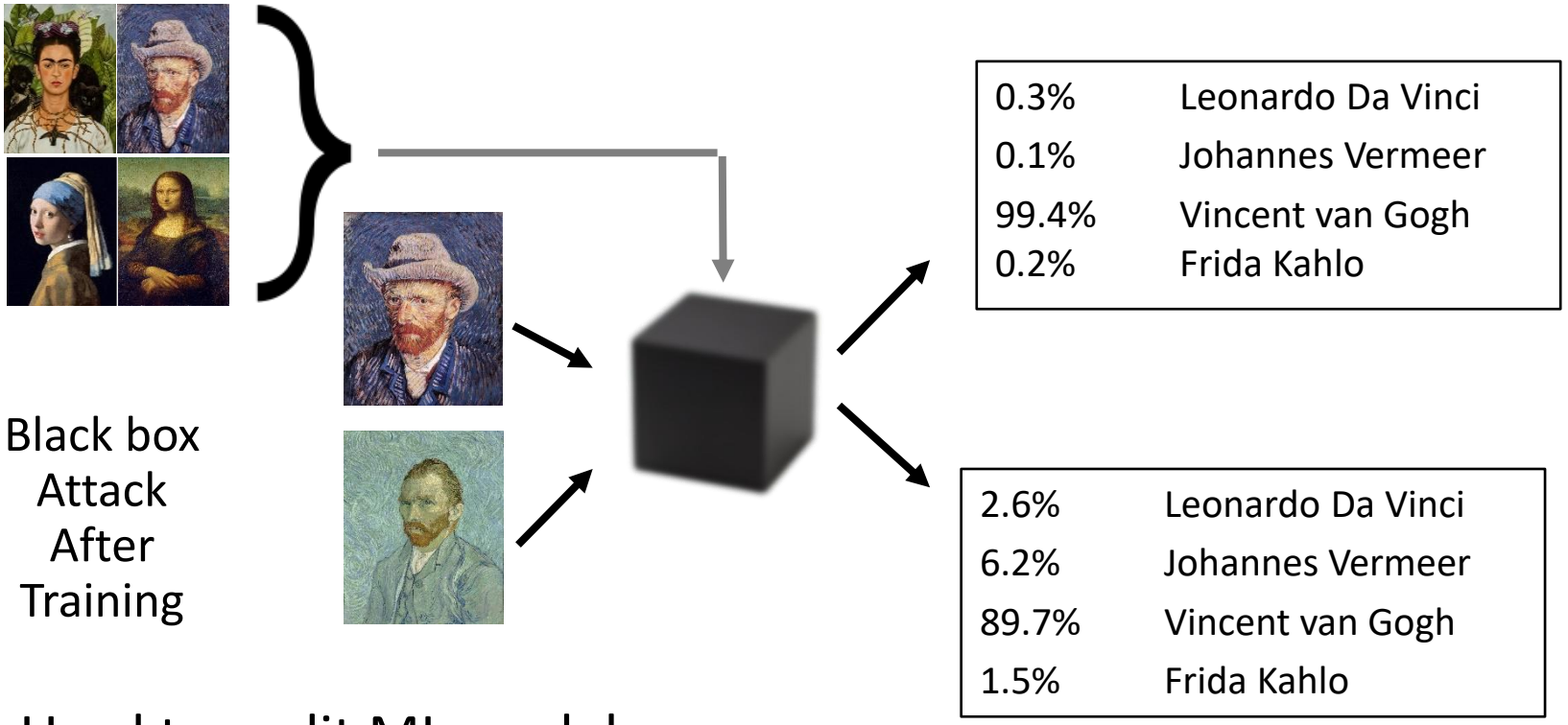
40.000 query
to steal model

White-box
Inversion
Attack

Local Inversion
has negligible cost

Membership Inference

- Leaks of one bit information about the overall training data.
- Machine learning models often behave differently on training sample versus a sample that they 'see' for the first time.

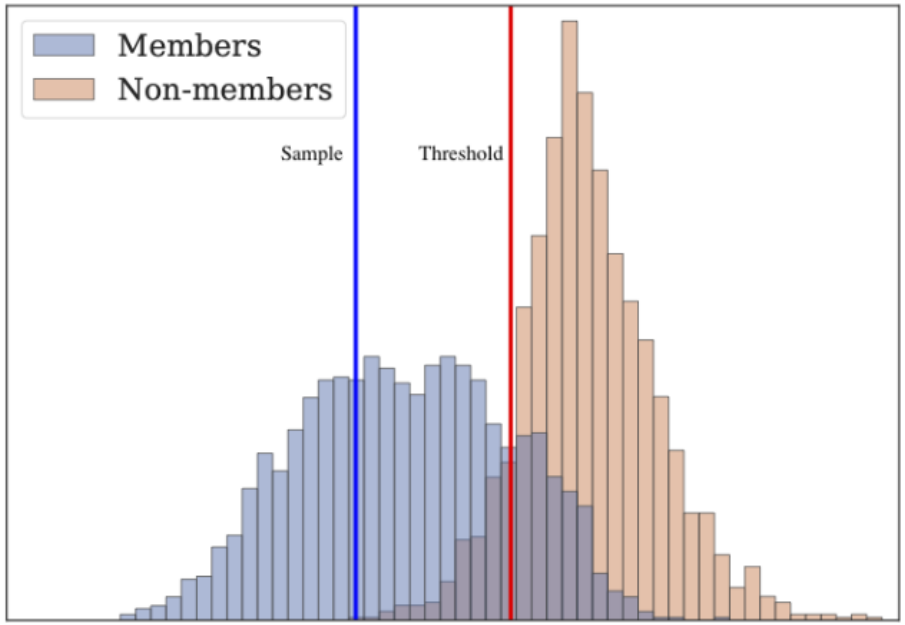


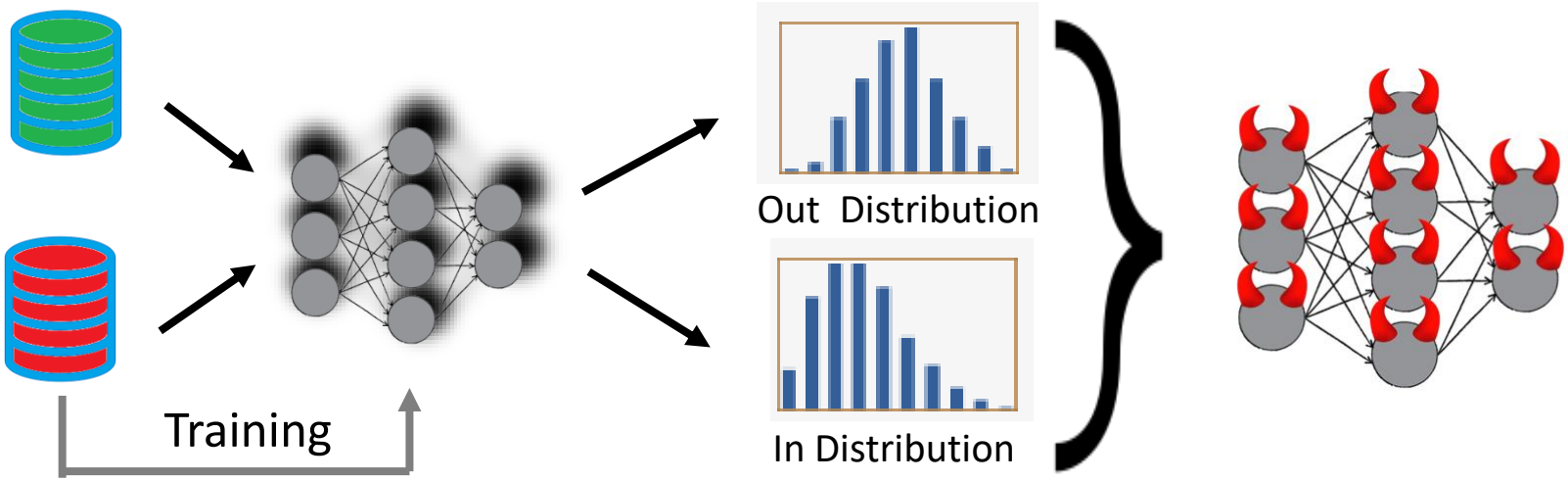
- Used to audit ML models.

Shadow Attack - LiRA

1. Obtain Shadow Dataset
2. Train Shadow Model(s)
3. Generate in/out distribution
4. Train Attack Model

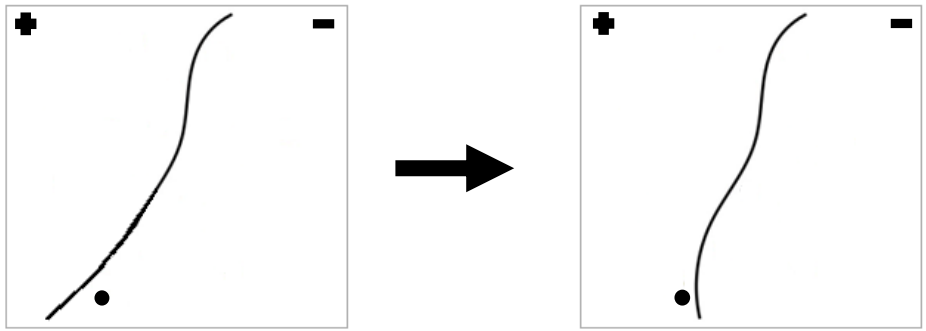
- Likelihood Ratio Attack

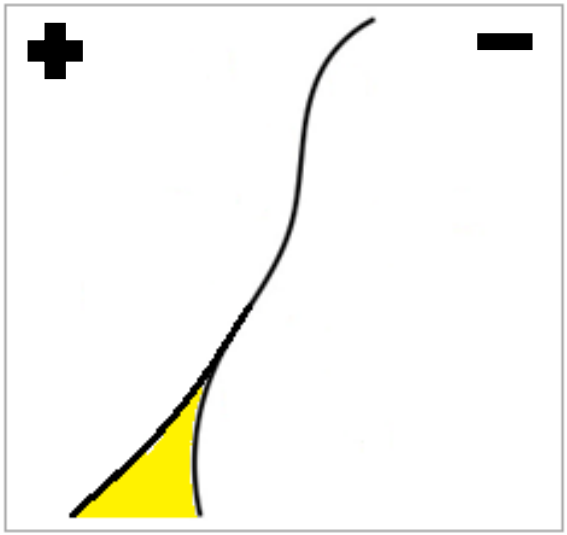




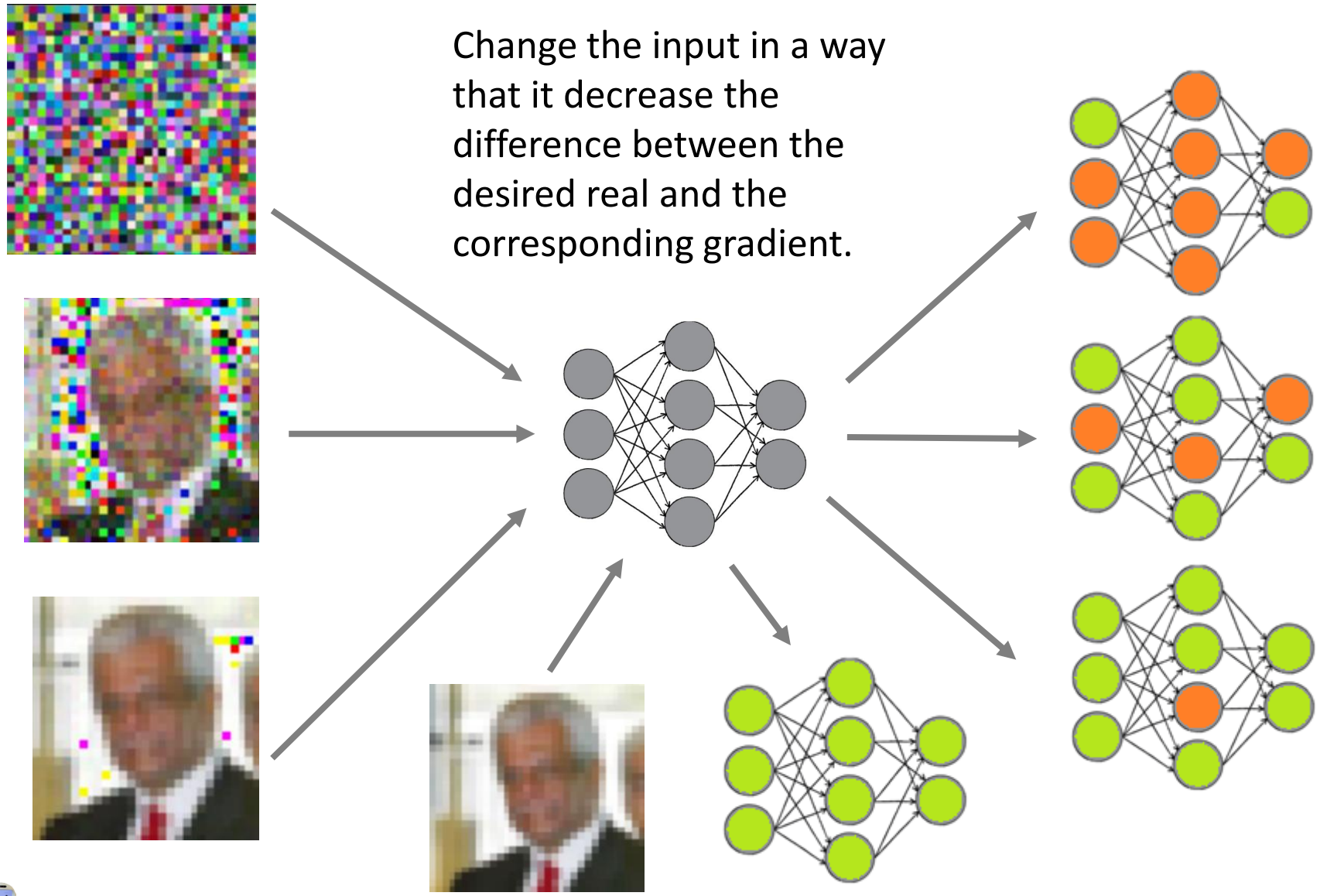
Reconstruction Attack

- Recovers “exact” training samples using the model updates.
- During training, the model is updated with the change corresponds to a data sample.
  - Data is given (•), model is given ( ).
  - Change is computed ( ).
- An attacker can swap what is given and what is computed.
  - Model is given ( ), change is given ( ).
  - Data is computed (•).





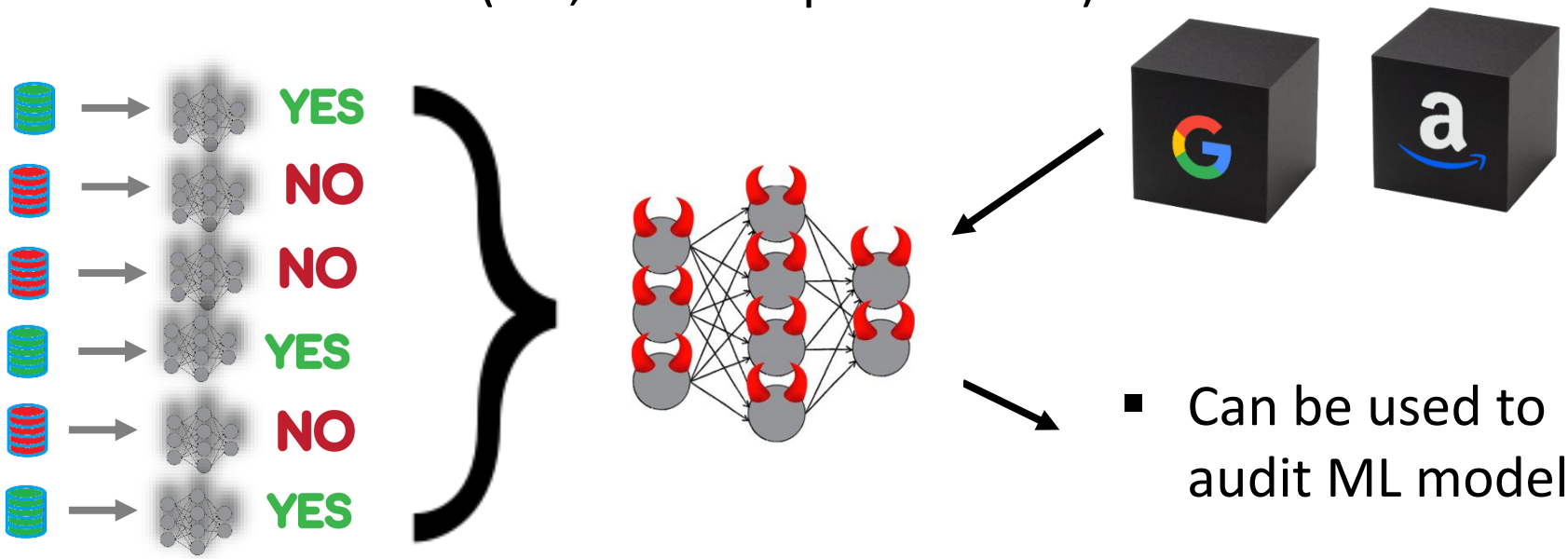
DLG - Deep Leakage from Gradient



Property Inference Attack

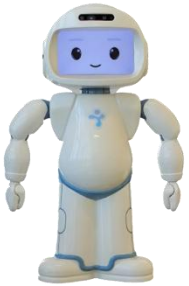
- Aim at uncovering properties of the dataset in which a given model was trained.
  - Unrelated to the main task of the model.
- Train many Shadow Models with and without property P and train the Attack Model on those models (i.e., on their parameters).

Black box Attack After Training



- Can be used to audit ML models.

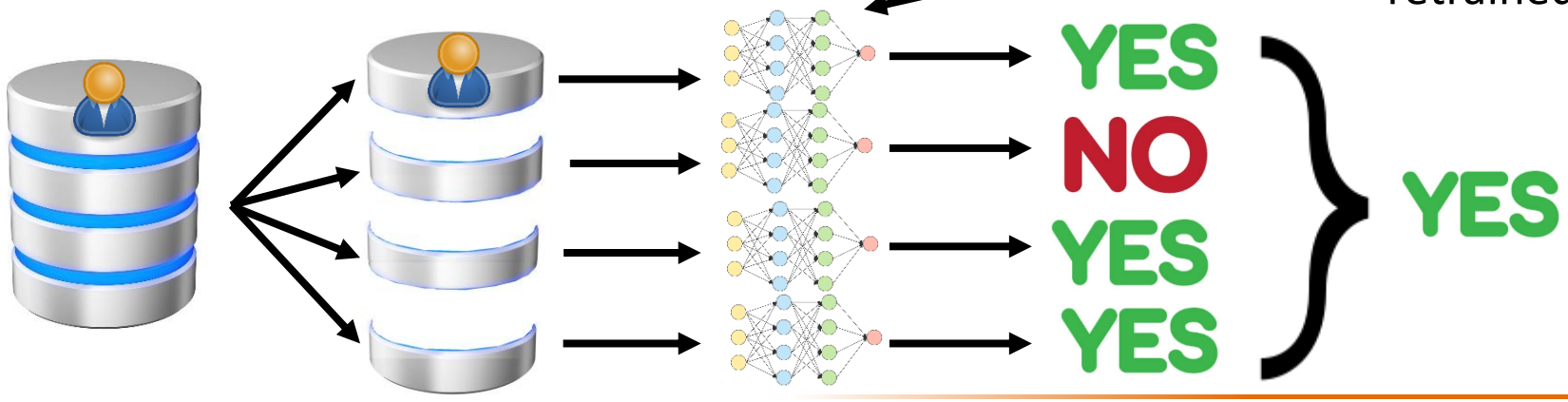
Defense Strategies



Machine Unlearning

- GDPR: „Right to be forgotten”
- Deleting and entry from a database is trivial.
- What about training data for ML?

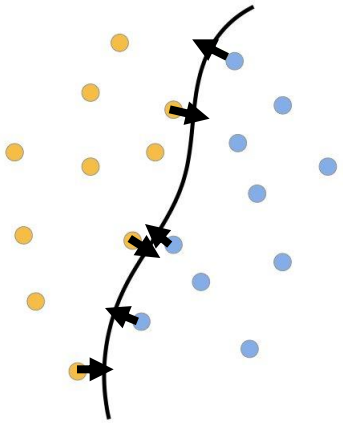
- Retrain from scratch cost too much in time and money.
- Alternative solution:
  - Slice training set and train independent models
  - Use majority vote for final prediction

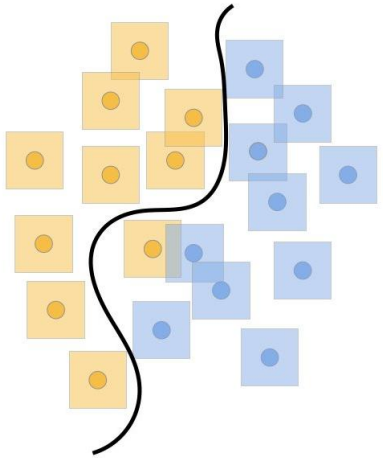


Robustness

- An ML is robust to adversarial examples if its output is insensitive to small changes to any plausible input that may be encountered in deployment.
  - Comes with trade-offs, i.e., accuracy drop.
- ML algorithm f is r-robust at x if f(x)=f(x+n) where n is in B.
  - Few large changes (p=1)
  - More small changes (p=2)

$$B_p(r) := \{\delta \in \mathbb{R}^n : ||\delta||_p \leq r\}$$





Adversarial Regularization

- An optimization method, which considers the attack accuracy.
- Attacker's Goal: For (x_0, y_0) find minimal n such that $f(x_0+n) \neq y_0$.
  - $g(x) = \text{Min}_n [\ n \mid f(x+n) \neq f(x)\ ]$
- Machine Learning's Objective: $\text{Min}_\theta [\ \text{Sum}_i(\ |f_\theta(x_i) - y_i|\ )\ ]$
- With Adversarial Regularization: $\text{Min}_\theta [\ \text{Sum}_i(\ |f_\theta(x_i) - y_i| + g(x_i)^{-1}\ ]$

- Pro
  - Easy to understand
  - Regularization is well studied and has other benefits
- Con
  - Reacts to a specific attack
  - Provides only empirical guarantee

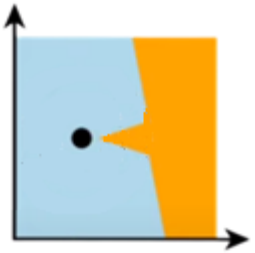
Certified Robustness

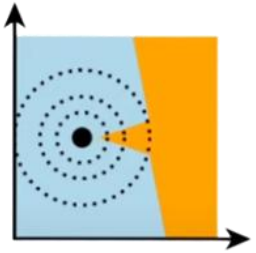
- Instead of arm-race between attacking and defending, design a model that are guaranteed to be robust to adversarial perturbations.
  - Idea is like in cryptography.
- Attacker: find any x and n pair such that f(x+n) is different from f(x).
  - Can be targeted, i.e., f(x+n) is given.
  - x can be prefixed too.

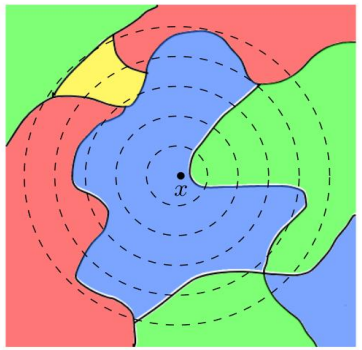
- Defender: prove that f is robust at x (e.g., check all n). For all x.
- Verification is hard, better to train f in such a way that robustness is guaranteed no matter what x is.

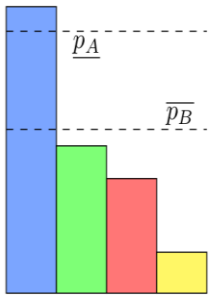
Randomized Smoothing

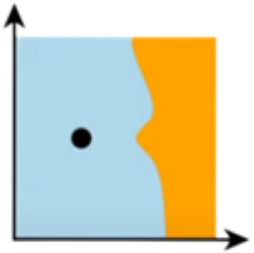
- Train the classifier h with already corrupted samples.
- Returns the most probable prediction with added random Gaussian noise.
- Run h(x+e) enough times and the most frequent class is the final (robust) prediction.
- Due to sampling, the ultimate guarantee (certificate) is probabilistic!
  - The strength of the robustness guarantee depends on the difference between the first and second most probably label.







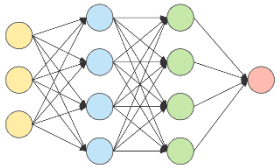


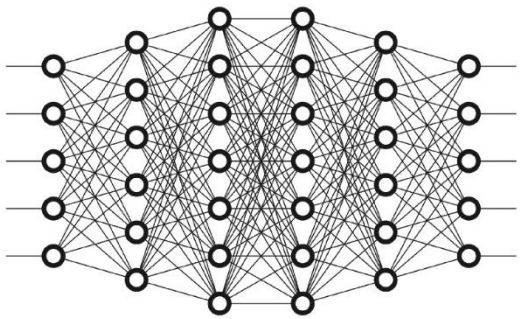


$$g(x) = \operatorname*{argmax}_y \mathbf{P}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[h(x+\epsilon) = y]$$

Empirical Privacy Defenses

- Pre-Processing
  - Remove Sensitive Data
  - Include Fake Data
- Post-Processing
  - Round probabilities
  - Only return predictions
- Before Training
  - Chose Adequate Model
  - Models Stacking / Ensemble / Distillation
- During Training
  - Gradient Normalization
  - Gradient Quantization
  - Gradient Sparsification
  - Gradient Truncation
  - Gradient Pruning
  - Gradient Regularization
  - Watermarking

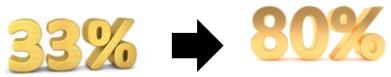


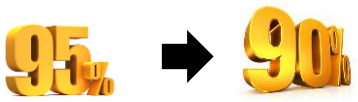


Provable Privacy Defense

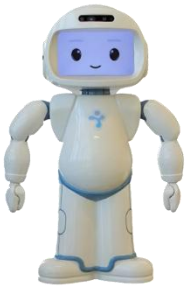
- Try to prevent the most fundamental privacy attack: MIA.
- The attacker knows that the actual database D can be either D_{in} or D_{out} depending on the presence or absence of their target.

- Attacker might have an initial suspicion about the database, i.e., $Pr[D = D_{in}]$ $(= 1 - Pr[D = D_{out}])$.
- How much information did the attacker gain by accessing the trained model / the training process?
  - Can be measured by the suspicion change.
- Differential Privacy ensures that the priori/posteriori change is not loo large.
  - Usually by injecting noise into the training process.





Trade Offs



Fairness

- Unfairness in data collection & labelling
  - Over / under representation of some groups of people (minorities).
  - Wierd: **W**estern, **I**ndustrial, **E**ducated, **R**ich, **D**emocratic
- Unfairness in feature selection
  - Accurate (but racist) classification
    - » Will pay back if 'Race' = 'Caucasian'
    - » Won't pay back otherwise
- Unfairness in defining target / class variable
  - Hiring based on predicted length of employment may disfavor women due to maternity.

- Fairness mechanism
  - Data Pre-Processing / Fair Training / Output Post-Processing

Fairness Definitions

- Unawareness
  - Disregard the sensitive attribute.
  - M(X) = M(X , S)
  - Corelated features still exists.
- Demographic Parity
  - The acceptance/rejection rates from the two groups must be equal.
  - $P_0(Y = y) = P_1(Y = y)$ for all $y \in \{ +, - \}$
  - Ignores any possible correlation between Y and S.
- Equalized Odds
  - For any label and attribute, a model predicts that label equally well for all values of that attribute.
  - $P_0(Z = z \mid Y = y) = P_1(Z = z \mid Y = y)$ for all y, z
  - It may not help closing the gap between two groups.

X	Non-Sens. Attributes
S	Sensitive Attribute
Y	Label
M	ML Model
Z	Model's Prediction

$P_0(\cdot) = P(\cdot \mid S = 0)$

$P_1(\cdot) = P(\cdot \mid S = 1)$

ML 3

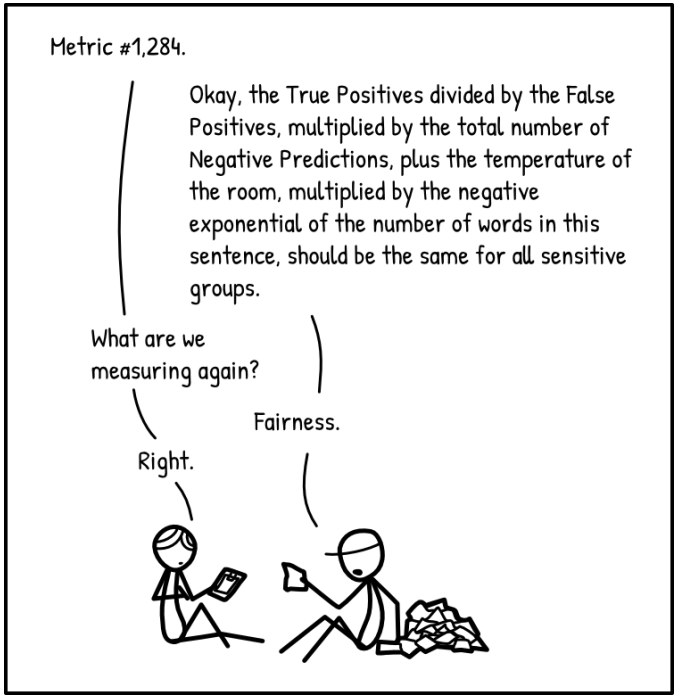
Impossibility Results

Costs

- Achieving Fairness
  - Over-sampling the minority
  - Under-sampling the majority
  - Weight the loss / Introduce regularization.
  - Etc.

Statistical parity
Group fairness
Demographic parity
Conditional statistical parity
Equal opportunity
Equalized odds
Conditional procedure accuracy equality
Disparate mistreatment
Balance for positive class
Balance for negative class
Predictive equality
Conditional use accuracy equality
Predictive parity
Calibration

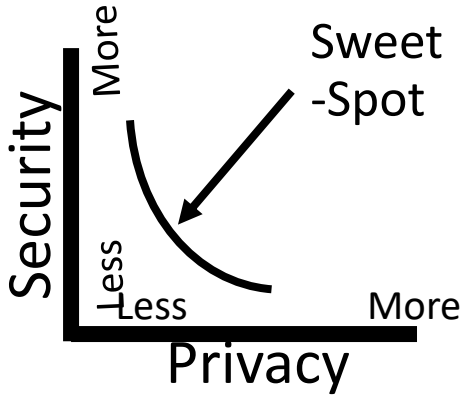
- Enforcing fairness …
  - results in accuracy drop.
- Imposing robustness …
  - results in accuracy drop.
- Applying privacy protection …
  - results in accuracy drop.

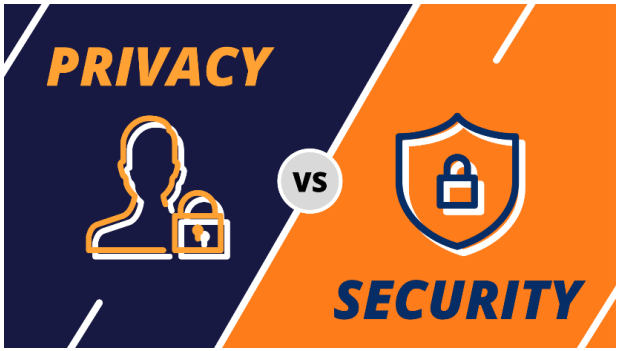


- How about ~~Accuracy~~ vs …?

Privacy vs Security

- The privacy-security tradeoff: finding a balance between robustness and privacy preservation.
  - Security mechanisms aim to detect malicious behavior.
  - Privacy preserving methods aim to hide individual specific information (training sample for ML, training set for FL).

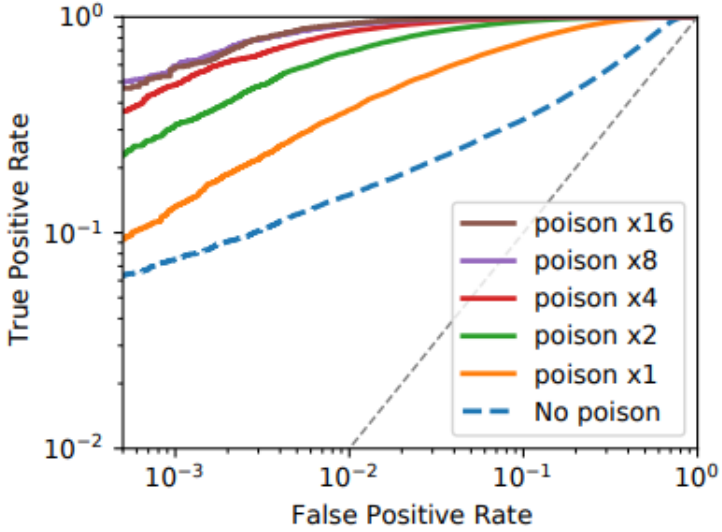


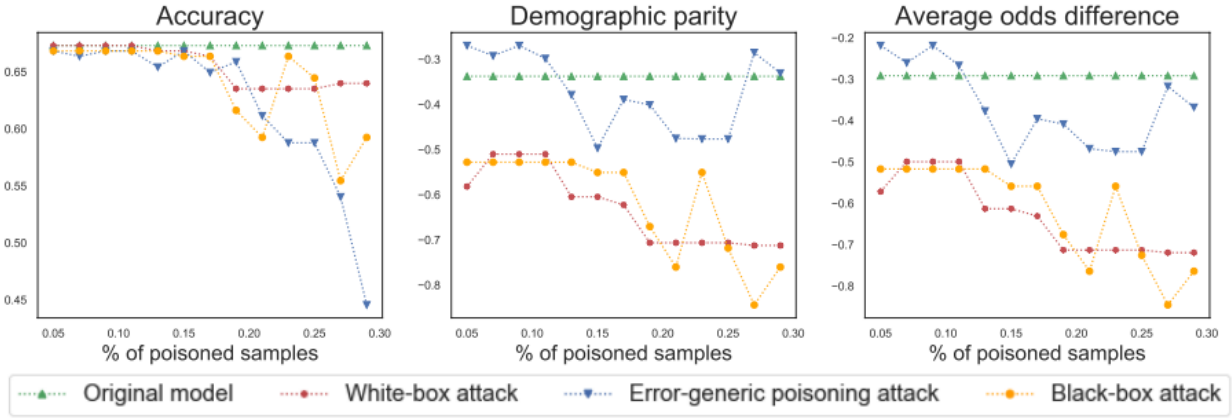


- Explainability vs …
  - Security: via poisoning one can effect the explanation map in training time.
  - Privacy: label is enough for a membership inference, so having access to more fine-grained information such as the confidence scores or explanations only increase its success.

Poisoning vs …

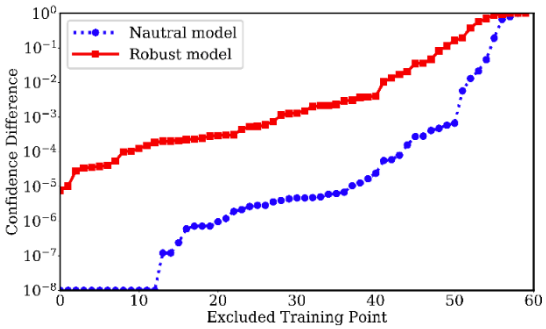
- Vs Privacy
  - Poisoning the training samples with the target sample but with incorrect label does increase the accuracy of Membership Inference Attack.
- Vs Fairness
  - Poisoning the training samples such that the fairness of the model decreases.



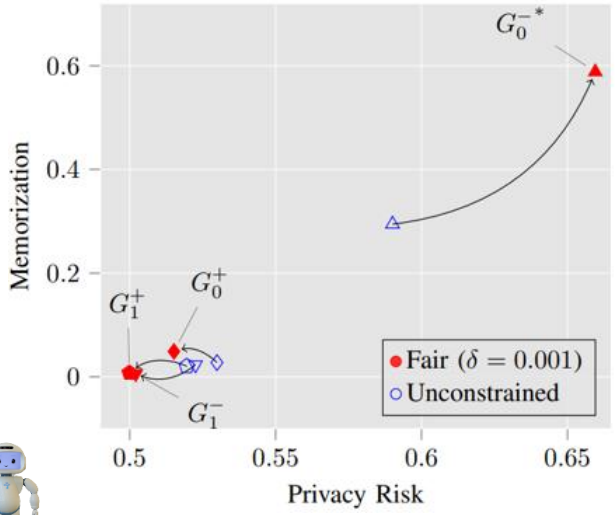


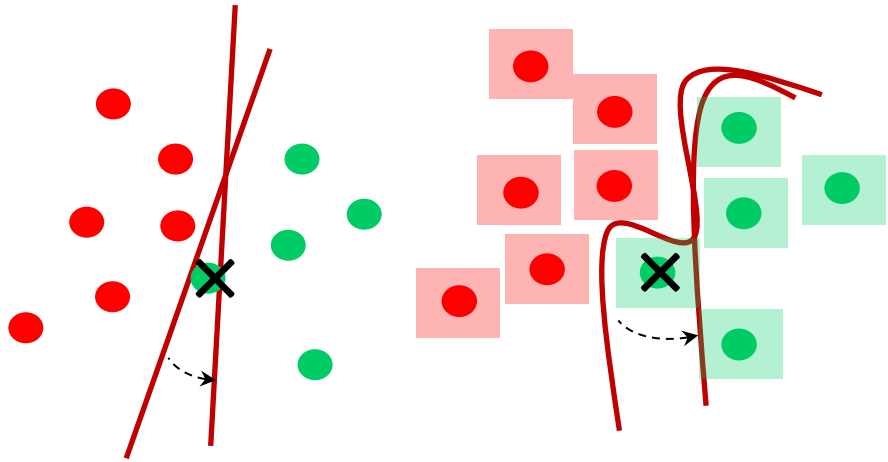
Privacy vs …

Change in output confidence with and w/o a training point.



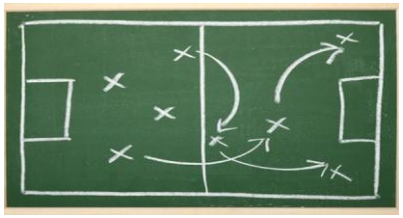
- Vs Robustness
  - A training sample can have larger impact on the decision boundary of a robust model.
  - Easier to launch Membership Inference Attack.
- Vs Fairness
  - Fairness constrains changes the influence of data points in a disproportionate way, i.e., it comes at the cost of privacy.
  - This effect is not distributed equally: the leakage increases on the unprivileged subgroups (which are the ones for whom we need fairness).



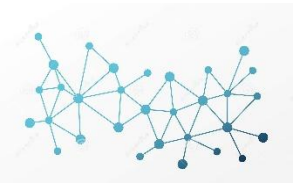


Take Away

- Machine Learning is not rocket-science!
  - Tricking trained ML models is called Evasion, and it could be done with adversarial examples.
  - Injecting unintended behavior into the ML model is called backdoors, and it can be done via poisoning.



- ML models could leak potentially sensitive data:
  - via Inversion the class average, via Extraction the model.
  - via Membership Inference the usage of a data, via Reconstruction the data.
- Attack could be defense too:
  - Machine Unlearning could remove a training data from the trained model.
  - Membership and Property inference could be used to audit ML models.
  - Adversarial Regularization against Adversarial Examples, Watermarking against model stealing, etc.



- Accuracy, Privacy, Security, Fairness, and Explainability are all connected and influence each other.

Control Questions

- What is an adversarial model, and what aspects does it have?
- What is the difference between

  model inversion,

  model extraction, and

  reconstruction attack?
- What is Machine Unlearning

  and Randomized Smoothing?

  How do they work, and

  what do they guarantee?

References

- Evasion Attacks against Machine Learning at Test Time
- Targeted Clean-Label Poisoning Attacks on Neural Networks
- Explanations can be manipulated and geometry is to blame
- Sponge Examples: Energy-Latency Attacks on Neural Networks
- Slowdown attacks on adaptive multi-exit neural network inference
- Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures
- Stealing Machine Learning Models via Prediction APIs
- Deep Leakage from Gradients
- Machine Unlearning
- Towards Deep Learning Models Resistant to Adversarial Attacks
- Certified Adversarial Robustness via Randomized Smoothing
- Entangled Watermarks as a Defense against Model Extraction