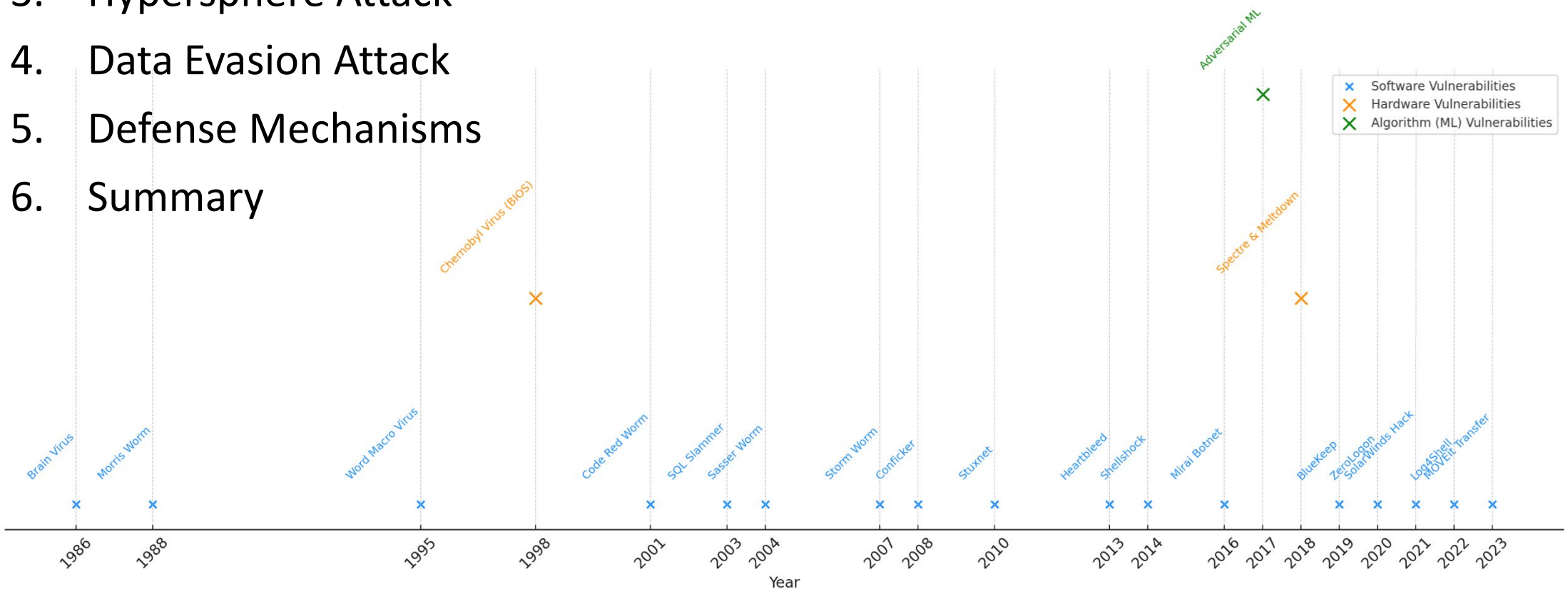
Securing Machine Learning: Evolving Threats, Attacks, and Defenses

Steve Wang

Outline

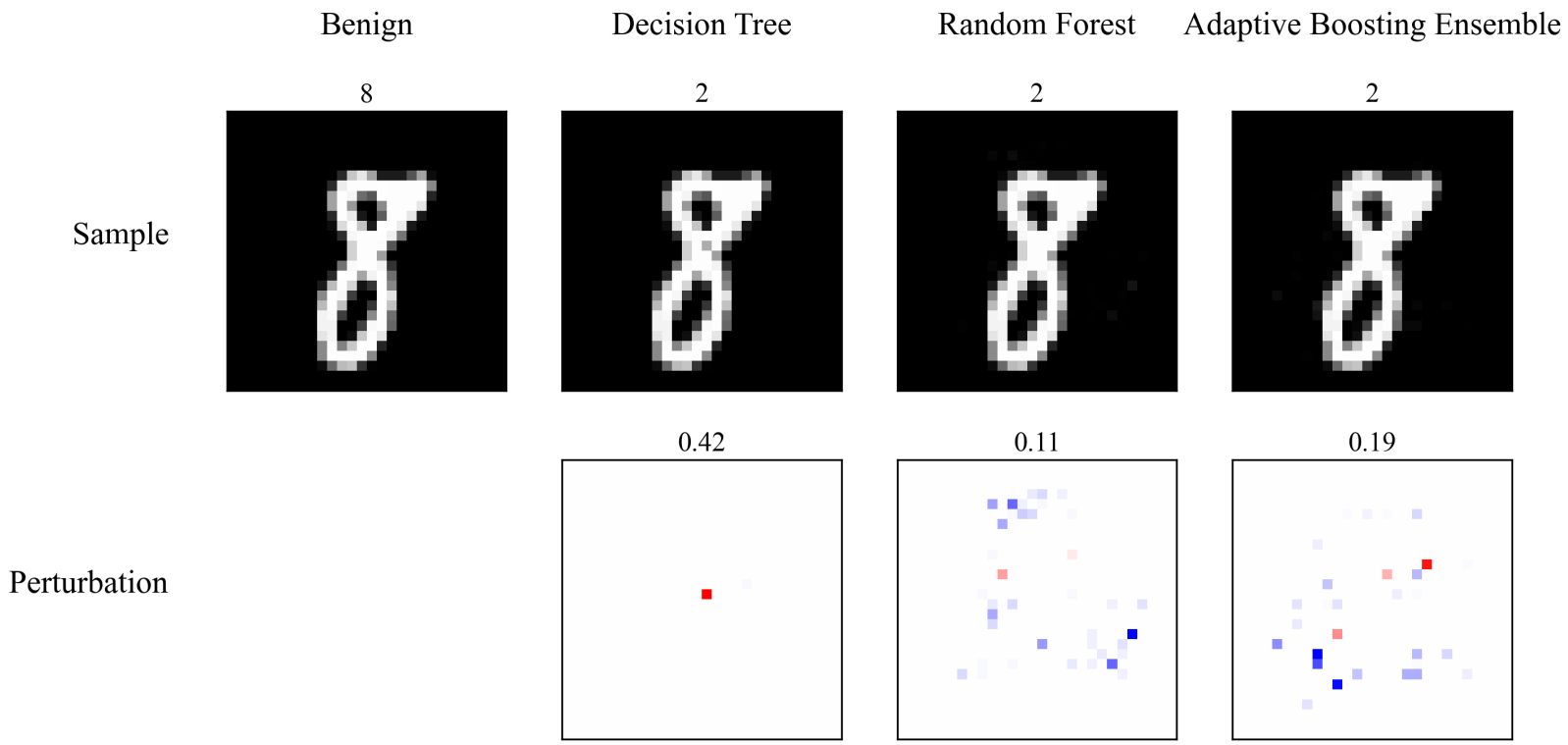
1. Introduction
2. Securing Machine Learning (ML)
3. Hypersphere Attack
4. Data Evasion Attack
5. Defense Mechanisms
6. Summary



Securing Machine Learning

- Machine learning (ML) has gathered increasing attention over recent years.
- The use of ML also raises attention to security in ML:
  - Tesla's Autopilot system for self-driving vehicles includes **48 neural networks** that make 1,000 predictions per timestep to autonomously operate motor vehicles.
  - In August of 2022, Sanford Health partnered with Dandelion Health, Inc. to "securely and ethically access representative, high-quality **clinical patient data**—including images, waveforms and structured health records—to enable the broader industry to build novel AI products, from aiding and automating medical decisions to improving diagnostics and drug development and more."
  - To fulfill the requirements for large amounts of data, some Large Language Models (LLMs) have resorted to using crowdsourced sources such as Wikipedia.

Adversarial Example 1



MNIST Handwritten Digits (Multiclass Classification):

- 784 features
- 60,000 training samples
- Decision Tree, Random Forest, AdaBoost

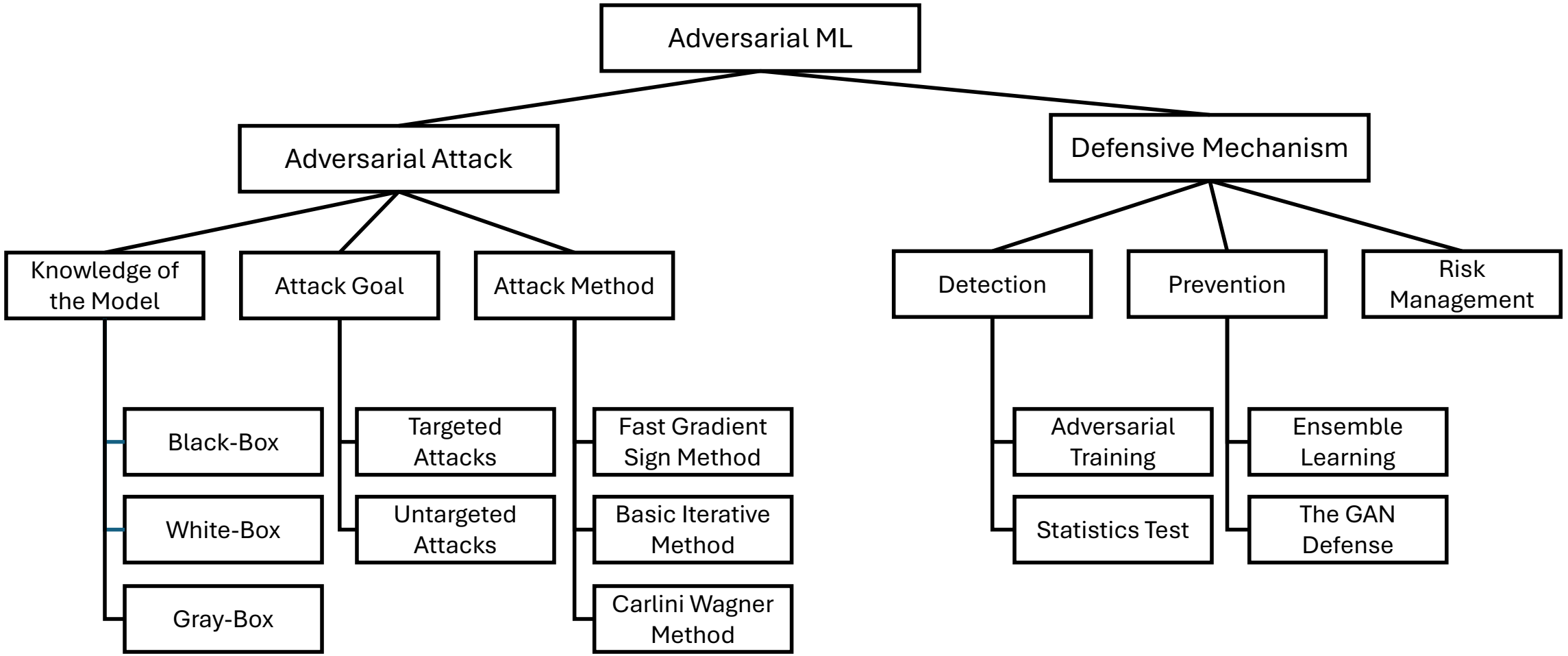
Testing environment:
Ubuntu Linux, Jupyter Notebook, matplotlib, numpy, pandas, scikit_learn, torchvision, ucimlrepo

Blue pixels indicate value-increasing perturbations, while red pixels indicate value-decreasing ones.

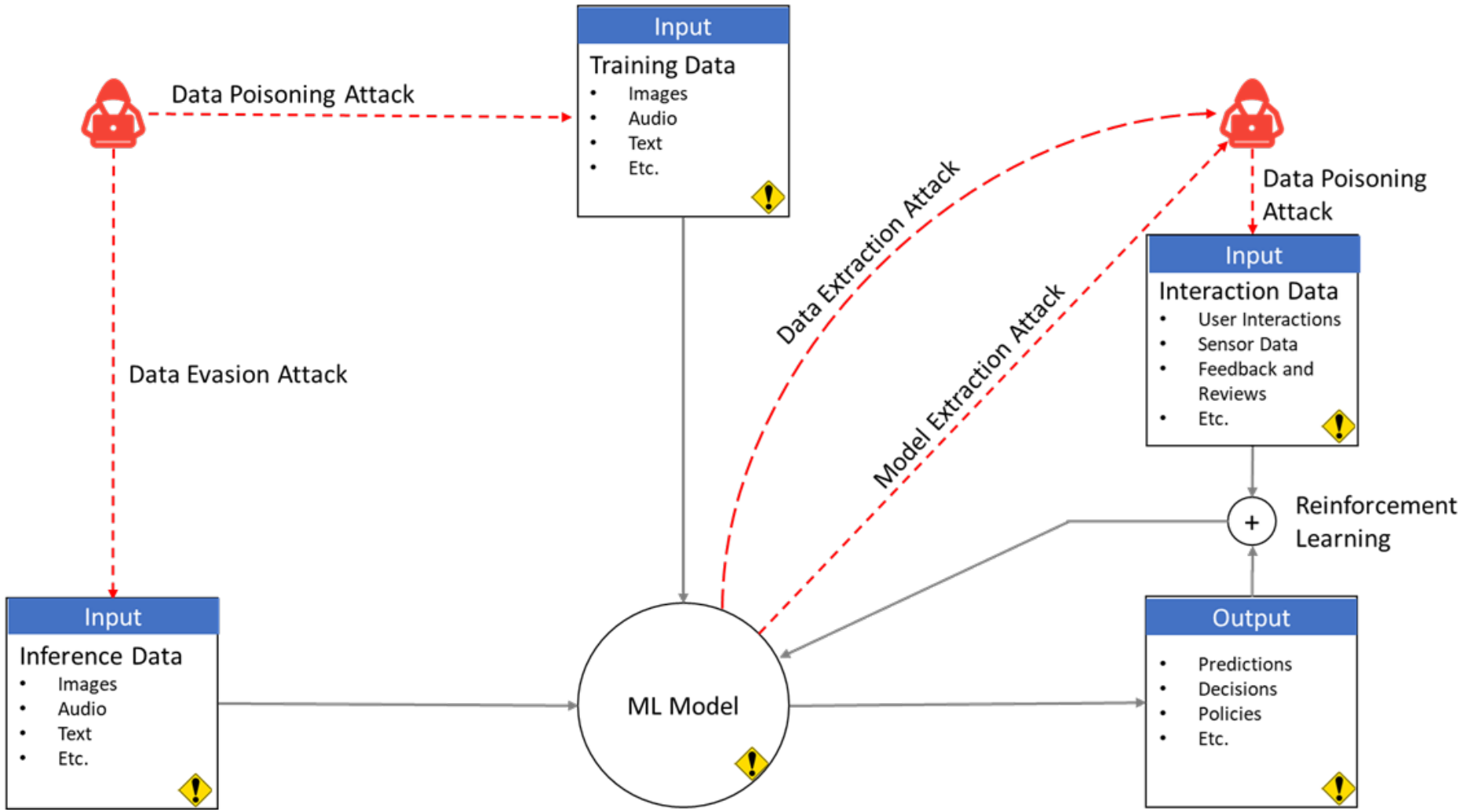
Adversarial Example 2

K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

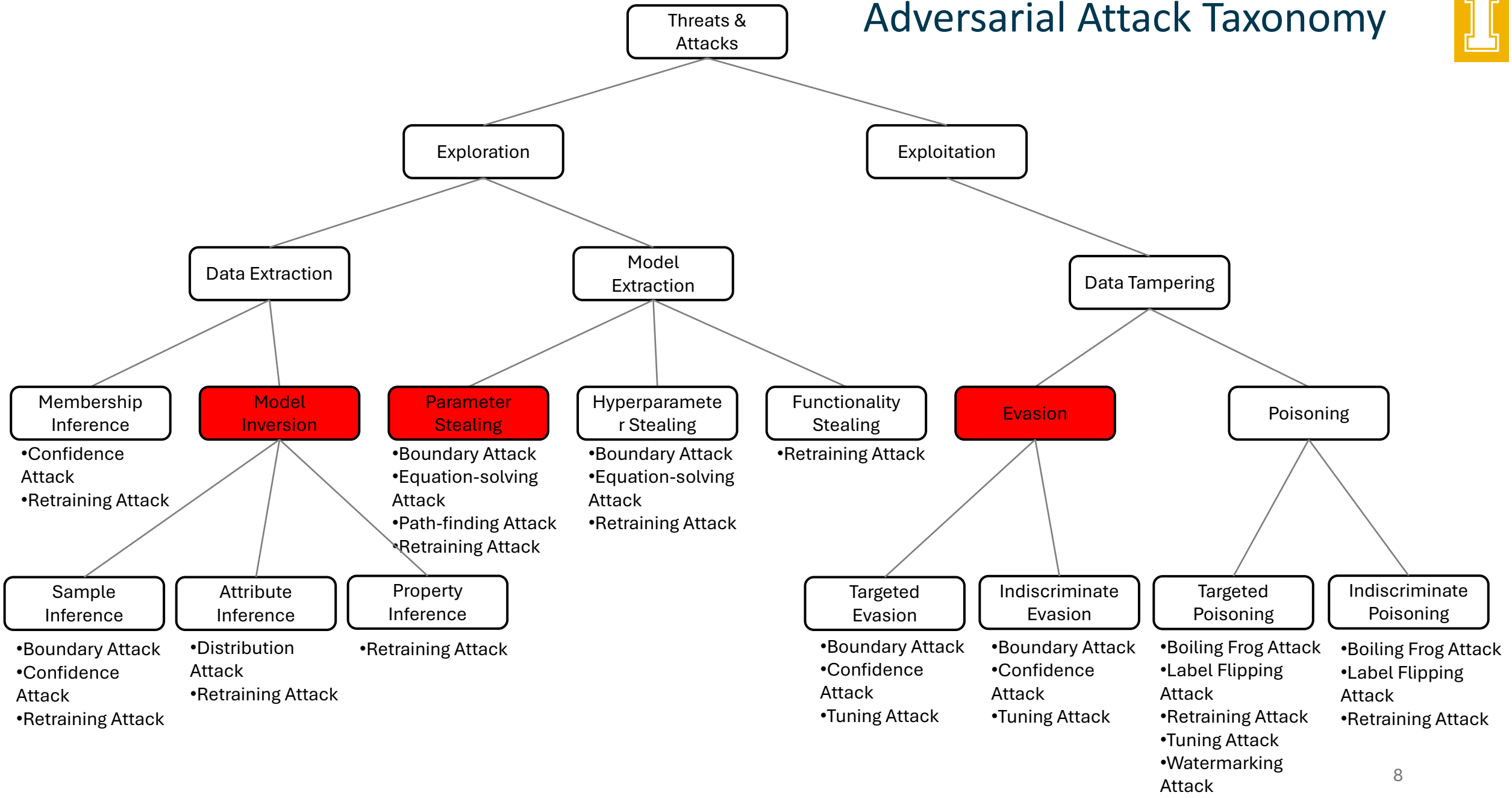
Adversarial ML investigates the security of ML methods against cyber-attacks.



ML Threat Model

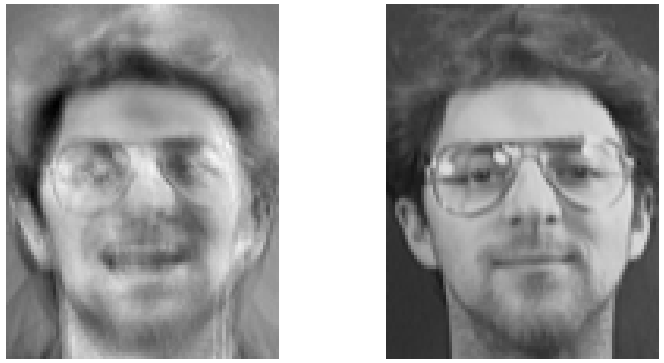


Adversarial Attack Taxonomy



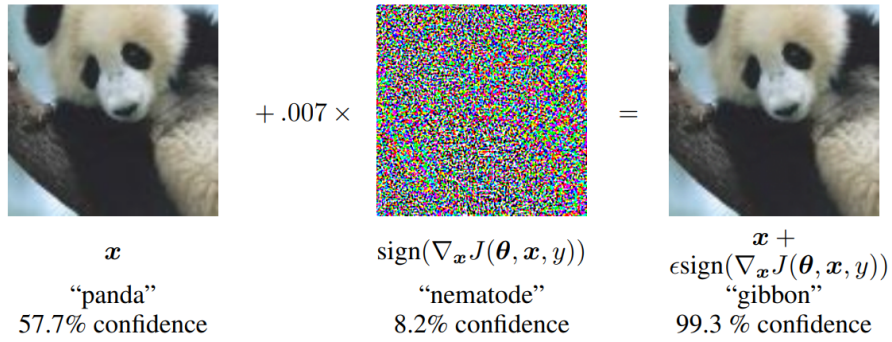
Threats to Machine Learning

- Exploratory attacks – data extraction and model extraction
  - Information is extracted that otherwise shouldn't be shared.



https://rist.tech.cornell.edu/papers/mi-ccs.pdf

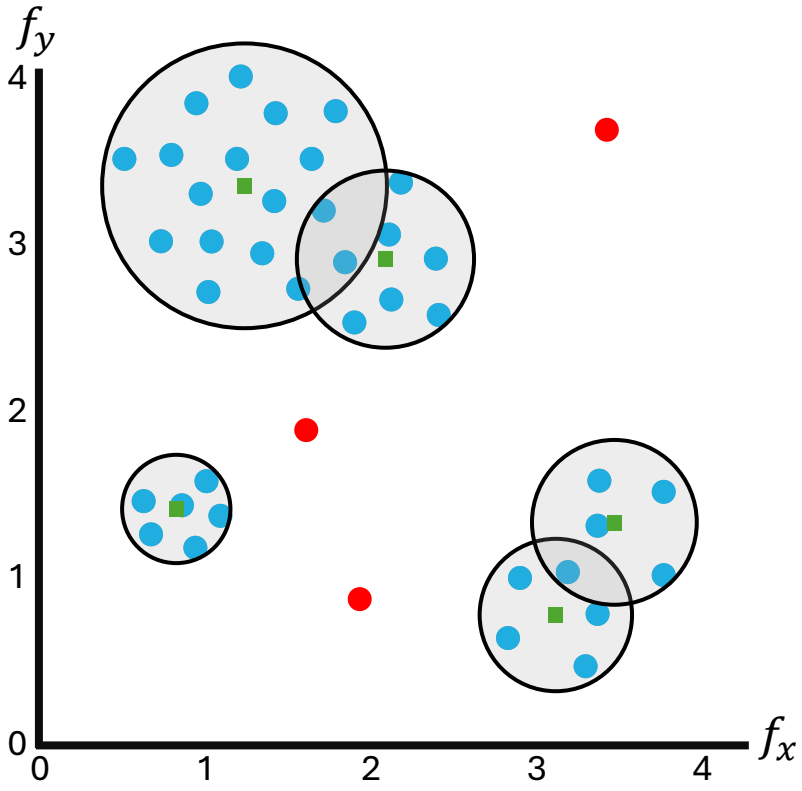
- Exploitative Attacks – data evasion and data poisoning
  - Some components of the ML system are modified to exhibit malicious behavior.



https://arxiv.org/pdf/1412.6572.pdf

Exploratory Attacks on Hypersphere-Based Models

- One-class classification
- A series of “hyperspheres” are fitted to the dataset.
  - Centroids chosen with K-Means, K-Medoids, etc.
  - Radii chosen with threshold criterion
- Unsupervised
- Interior points are labeled as 1
- Exterior points are labeled as 0

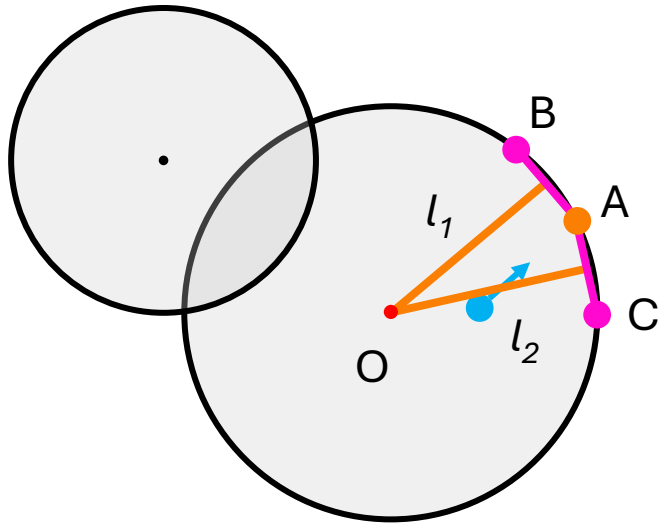


Hypersphere Attack

a black-box attack

1. Reflection

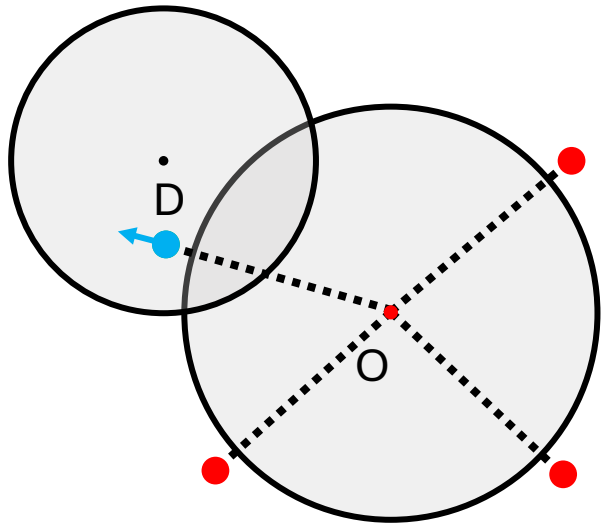
- Given an initial point in a hypersphere as well as a random direction, travel in the direction until the decision boundary point is found.
- Identify additional, close boundary points.
- Calculate the intersection of the perpendicular bisectors of the lines that form from the boundary points to find the centroid .
- Calculate the distance between the captured centroid and a boundary point to find the radius r.



Hypersphere Attack

2. Infection

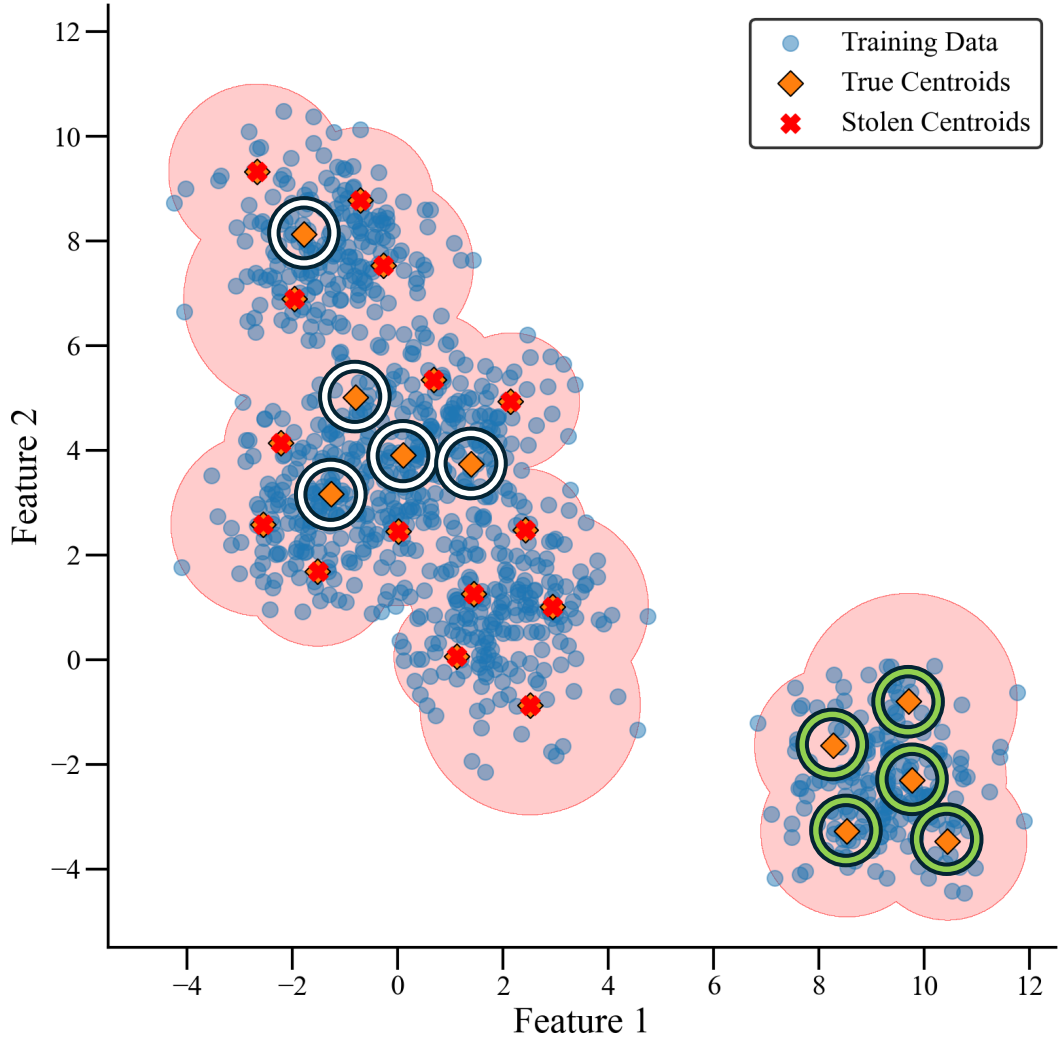
- Query a series of points that lie a distance of $r + \varepsilon$ from the captured centroid where ε is small.
- For every query point that lies outside of any known hyperspheres but is classified as an interior point by the model, repeat the reflection portion of the attack with the point and the direction from the centroid to the point.



Hypersphere Attack Evaluation

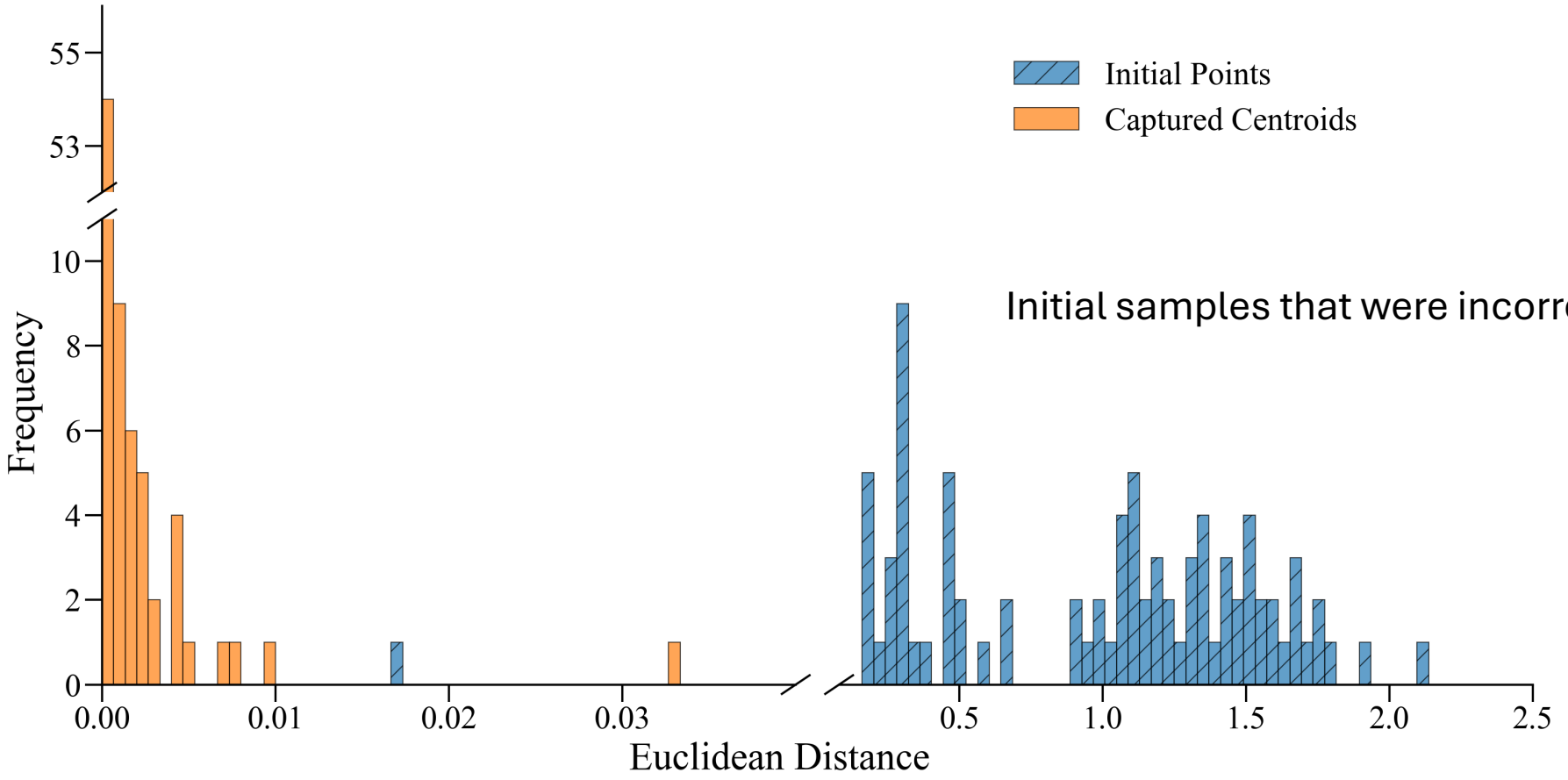
- Randomized (Infection – Parameter Stealing)
  - 2 features
  - 1,000 training samples
  - 25 Hyperspheres
- RT-IoT2022 (Reflection – Parameter Stealing)
  - 81 features
  - 9,266 training samples
  - 350 Hyperspheres
- UCI Digits (Reflection – Model Inversion)
  - 64 features
  - 142 training samples
  - 3 Hyperspheres

Evaluation Results - Randomized



- White-circle centroids **overlap** with other hyperspheres.
- Green-circle centroids are **disconnected** from the initial attack region.
  - They will be captured with sufficient scan points.

Evaluation Results – RT-IoT2022

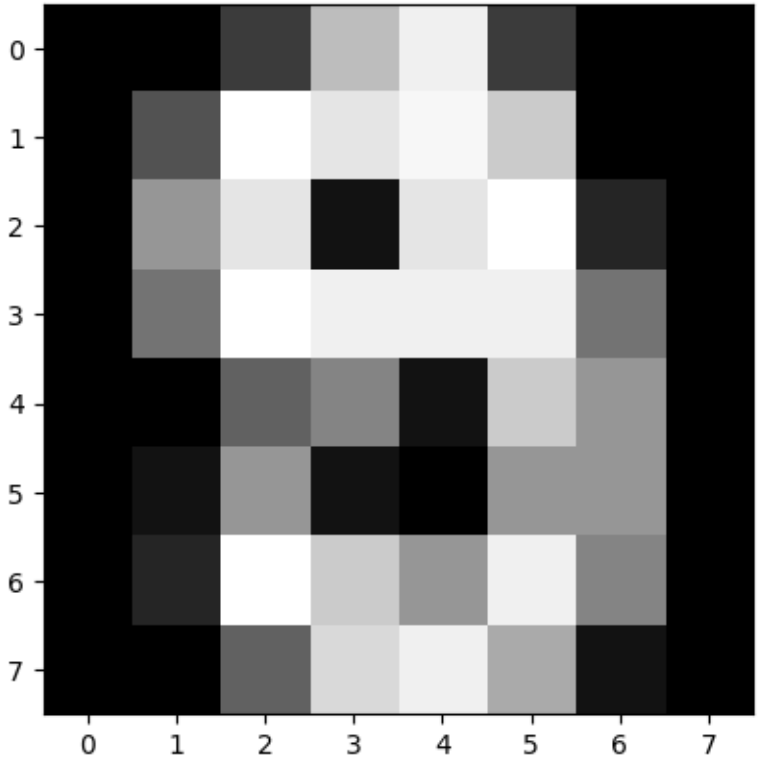


Initial samples that were incorrectly predicted.

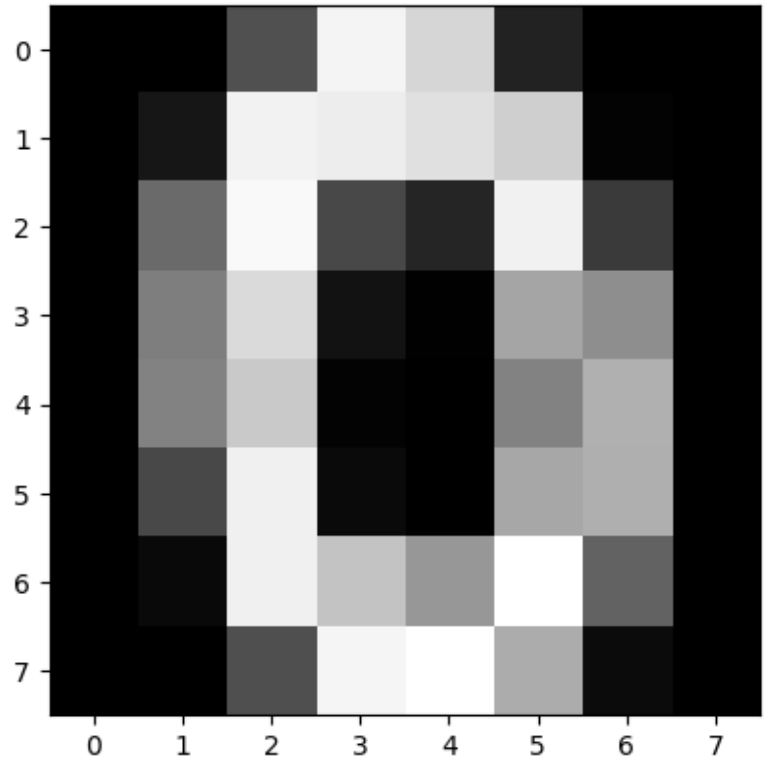
Comparison of distances between initial samples and true centroids with captured centroids and true centroids.

Evaluation Results – Digits

The proposed attack can also extract meaningful information (model inversion) from the training dataset of a targeted model.



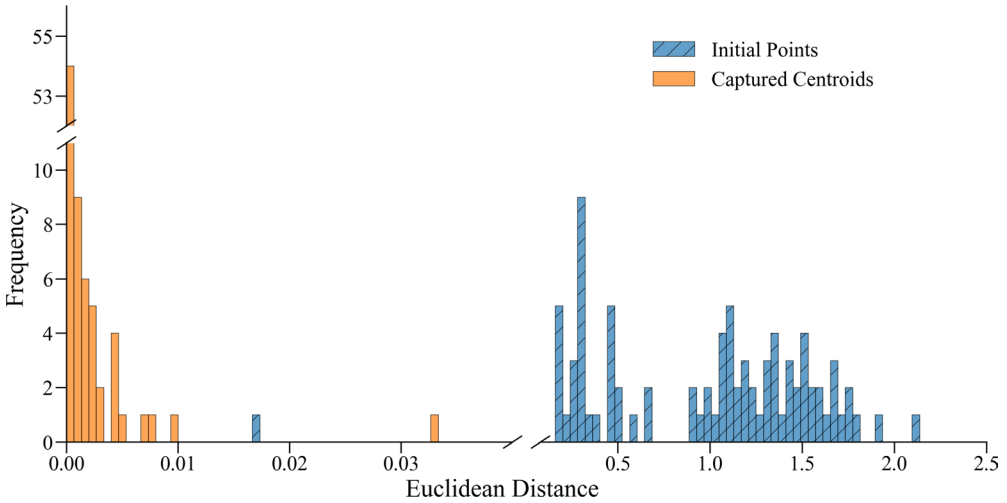
Initial sample (9) used for the attack on the Digits dataset.



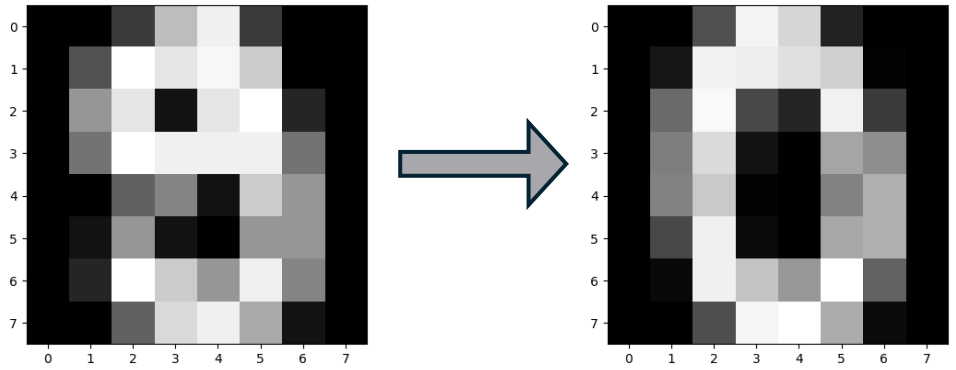
Capture digit (0) from attack on the Digits dataset.

Hypersphere Attack

- Hypersphere attack can act both as a parameter stealing attack and a model inversion.
- Requires hard-label black box outputs.



Parameter Stealing



Model Inversion

Exploitative Attacks Targeting Tree-Based ML Models

- Data evasion attacks
  - Manipulate input data
  - Cause a model to make incorrect predictions or classifications

$$M(x) = o$$

Targeted attack $\underset{\delta}{argmin}\|\delta\| \; s.t. \; M(x+\delta) = \bar{o}, \bar{o} \neq o$

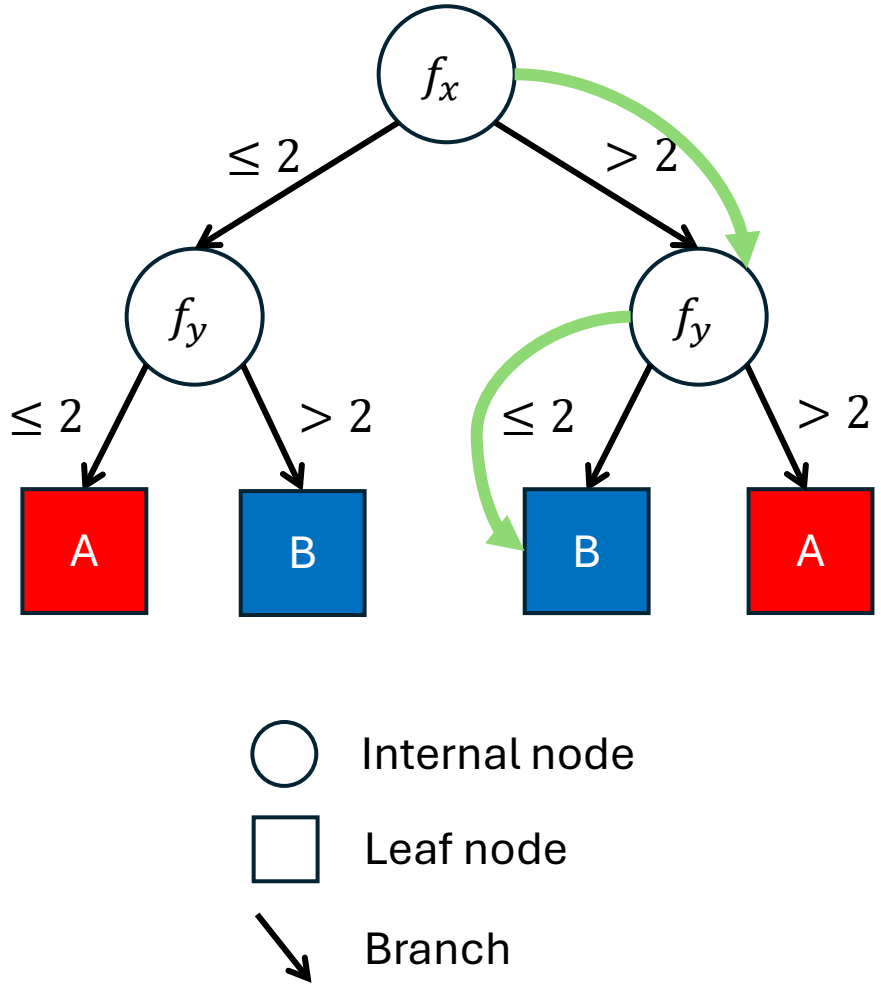
Untargeted attack $\underset{\delta}{argmin}\|\delta\| \; s.t. \; M(x+\delta) \neq o$

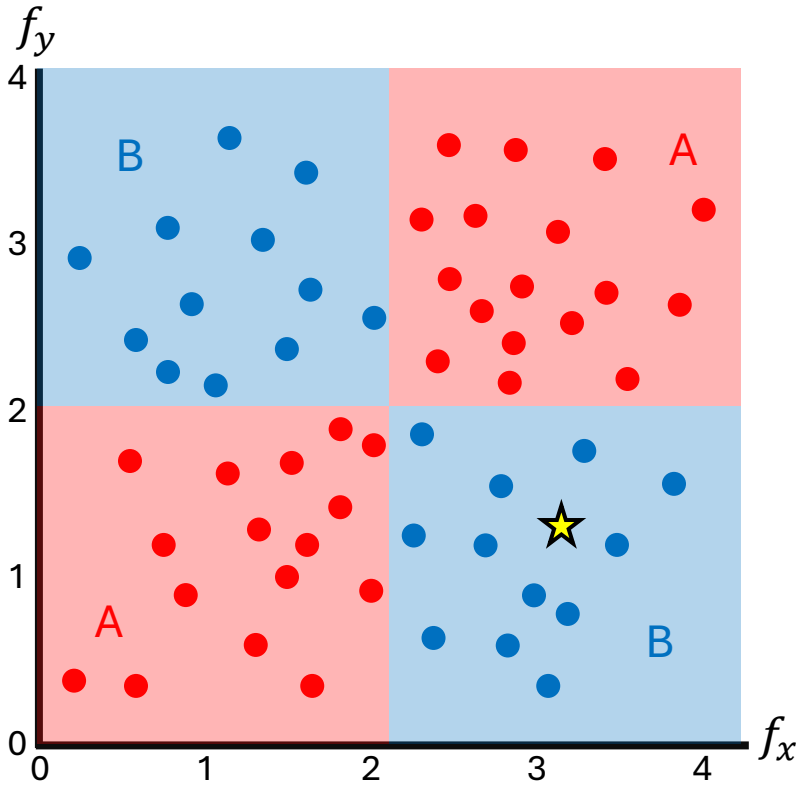
$$x = (x_1, x_2, \ldots, x_n)$$

- Data evasion attacks
  - Maximize “**confusion**”
  - Minimize “**perturbation**”

Euclidean distance $\delta = \|\bar{x} - x\| = \sqrt{\sum_{i=1}^{n} (\bar{x}_i - x_i)^2}$

Decision Trees

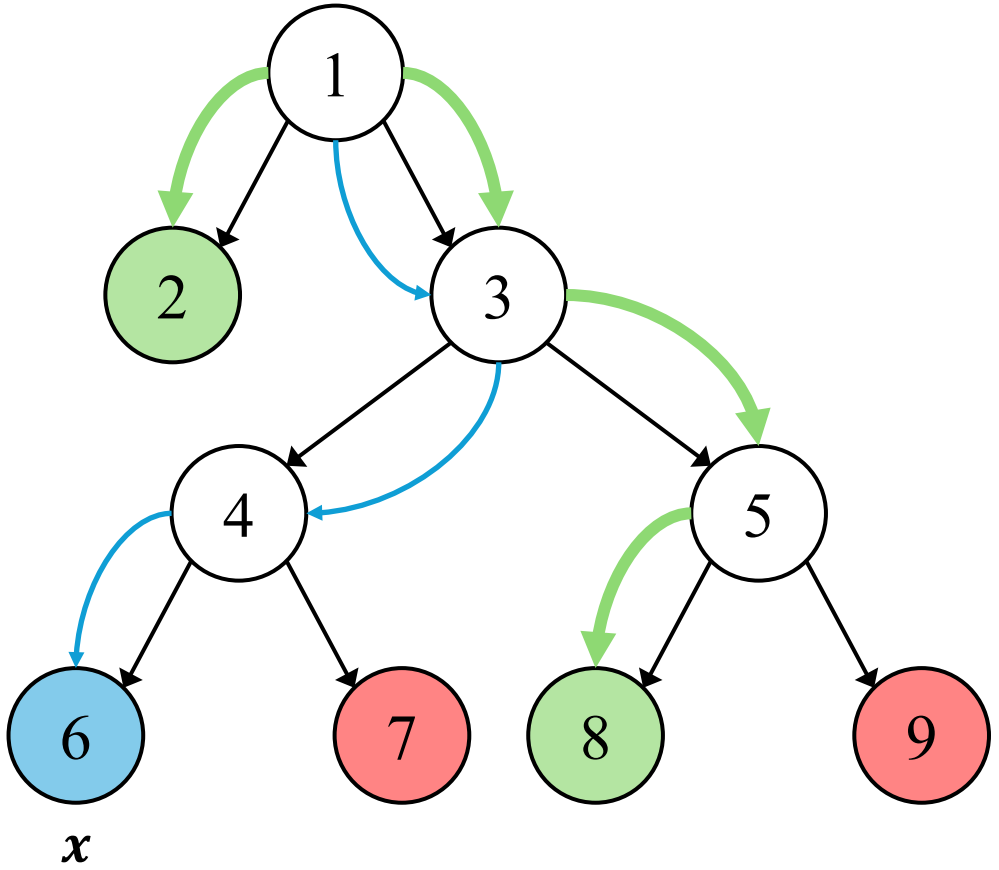




Single Tree Attack

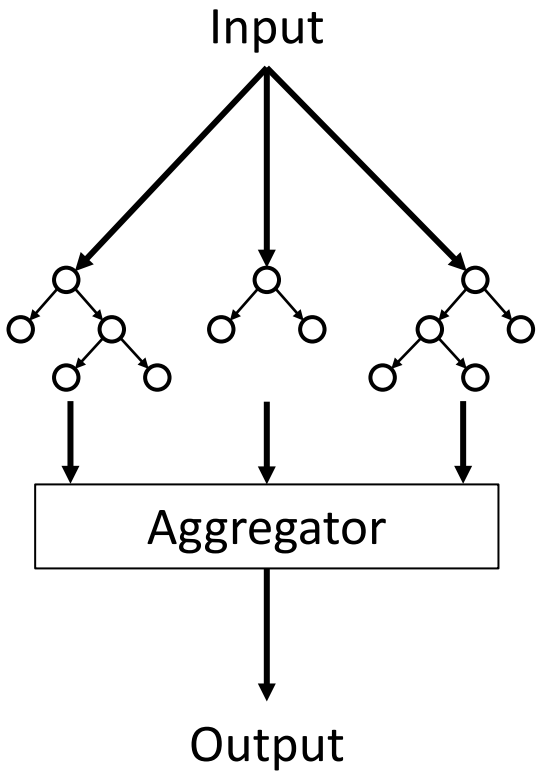
1. Find path for the starting node.
2. Find paths for target nodes.
3. Calculate deviated paths between the target and starting paths.
4. Determine conditions for following deviated paths (resulting in perturbations).
5. Choose the optimal perturbation.

$$argmin_{x \in \bar{X}} \|\bar{x} - x\|$$

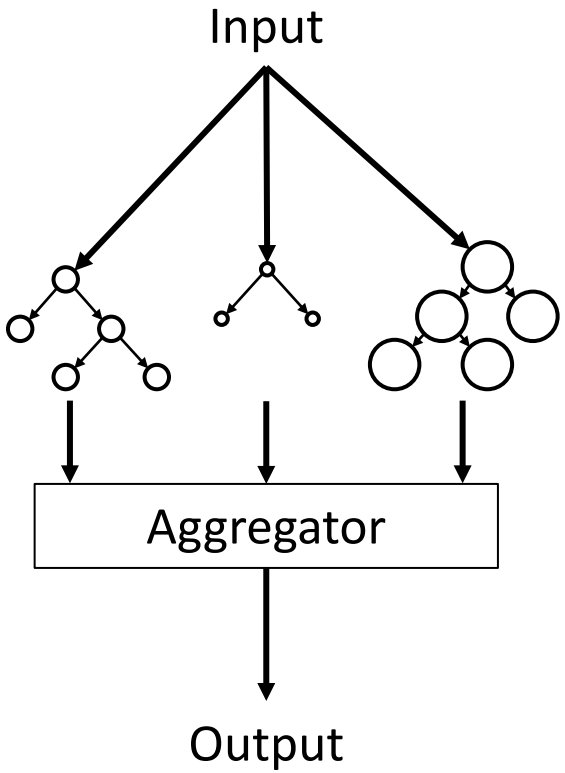


Tree-ensemble ML Models

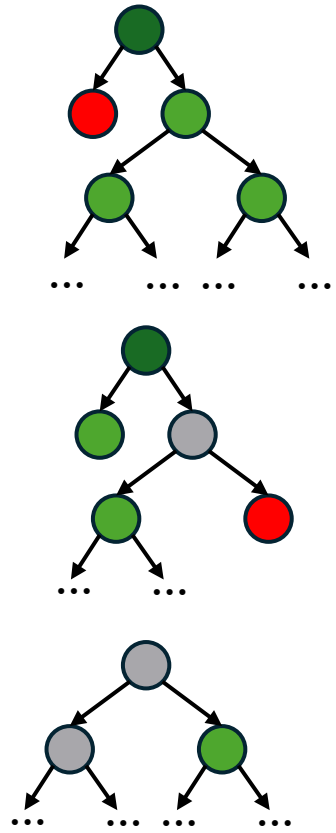
Supervised Tree Ensembles – Bagging, e.g., Random Forest



Supervised Tree Ensembles – Boosting, e.g., Adaptive Boosting

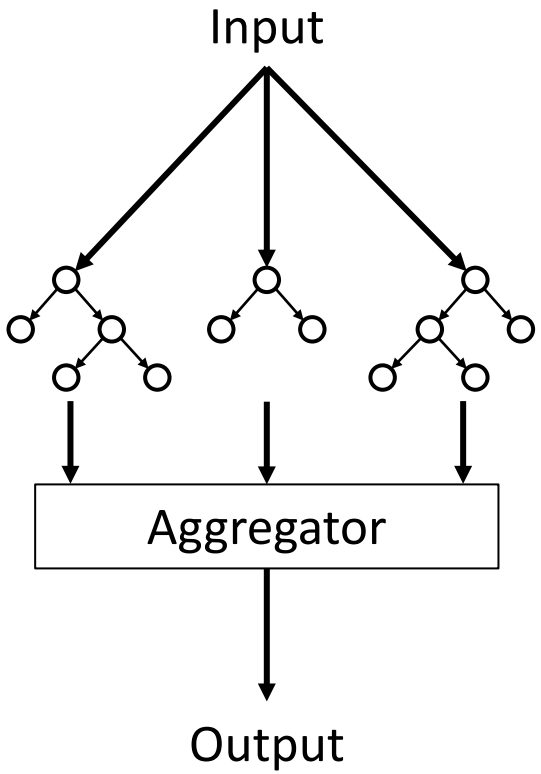


Unsupervised Tree Ensembles – Isolation Forest

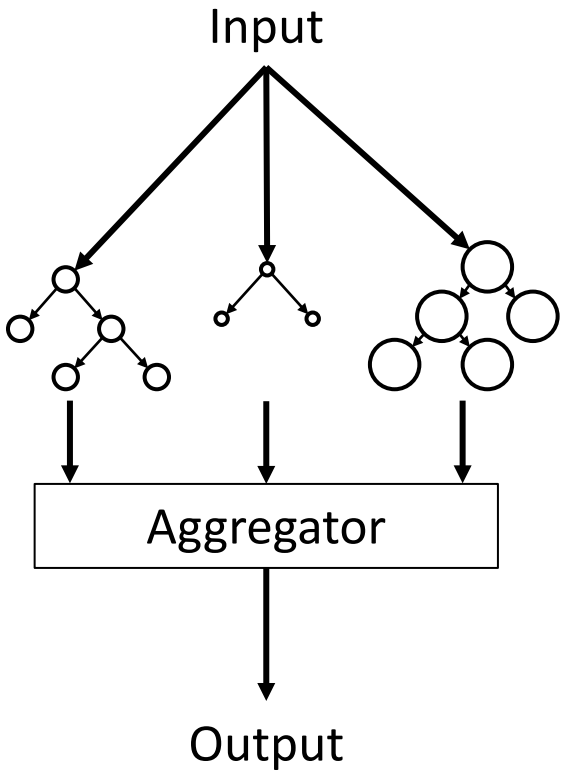


Supervised Tree Ensembles

Supervised Tree Ensembles – Bagging, e.g., Random Forest



Supervised Tree Ensembles – Boosting, e.g., Adaptive Boosting



Max "confusion", min "perturbation"

A scaled damage metric s

$$s(x, \bar{x}, E, t, \bar{o}) = (d(\bar{x}, E, \bar{o}) - t)^2 + (\|\bar{x} - x\|)^2$$

$$d(\bar{x}, E, \bar{o}) = \frac{1}{|E|} \sum_{T \in E} T_{\bar{o}}(\bar{x})$$

$|E|$: number of trees in the ensemble

$t = \frac{1}{c}$, c is the number of classes

Unsupervised Tree Ensembles

Leaf nodes do not assign labels. A sample is generated for each leaf node in the tree during the unsupervised attack.

Max "confusion", min "perturbation"

A scaled damage metric s

$$s(x, \bar{x}, E, t, \bar{o}, n) = (-\log_2 t \cdot c(n) - E(\bar{x}))^2 + (\|\bar{x} - x\|)^2$$

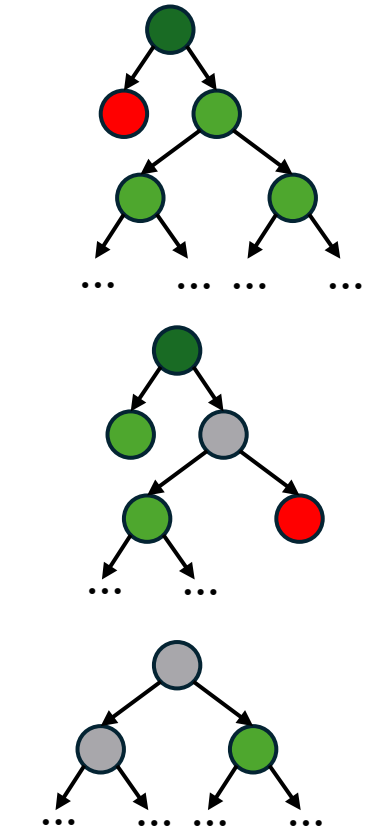
$c(n)$: is the average unsuccesfful search

$E(x)$: is the average height of the leaf node that contain x across all trees in the ensemble.

t: a threshold hyperparameter

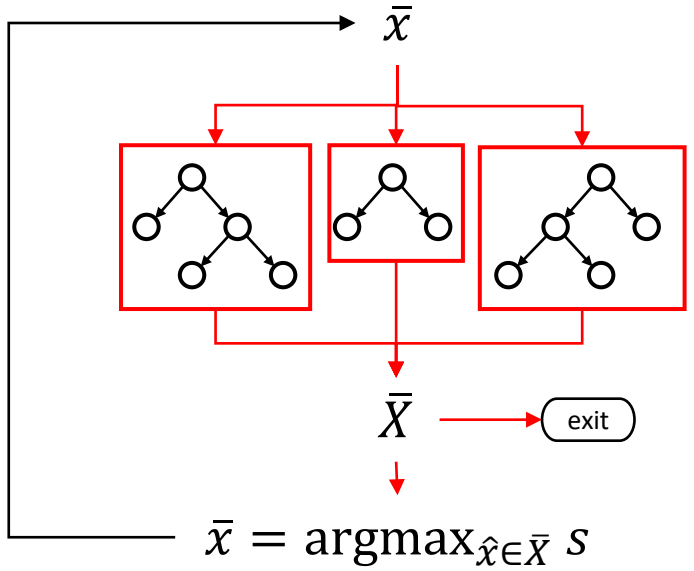
Anomaly scoring function: $a(x, n, t) = 2^{-\frac{E(x)}{c(n)}} - t$

Unsupervised Tree Ensembles – Isolation Forest



Ensemble Tree-based Attack

1. Evade each tree separately using $\bar{x}$ and store resulting evading candidates in $\bar{X}$.
2. If there is $\bar{x} \in \bar{X}$, such that $E(\bar{x}) = \bar{o}$, return $\bar{x}$.
3. Update $\bar{x}$ to max confusion and minimize perturbation.
4. Repeat until $\bar{x}$ successfully evades target model.



Evasion Attacks

Algorithm 1 Attack on a single tree-based learner

Input
  T Fitted single tree-based target model
  $\mathbf{x}$ Initial benign sample
  $\bar{o}$ Target output $\neq T(\mathbf{x})$
Output
  $\bar{\mathbf{x}}$ Evading sample
$L \leftarrow$ leaf nodes assigning label $\bar{o}$ in T
$\bar{\mathbf{X}} \leftarrow [\,]$
for $l \in L$ **do**
  $\bar{\mathbf{x}} \leftarrow \mathbf{x}$
  modify $\bar{\mathbf{x}}$ to fulfill criteria from root node of T to l
  append $\bar{\mathbf{x}}$ to $\bar{\mathbf{X}}$
end for
return $\arg\min_{\bar{\mathbf{x}} \in \bar{\mathbf{X}}} \|\bar{\mathbf{x}} - \mathbf{x}\|$

Algorithm 2 Attack on ensemble tree-based learner

Input
  E Fitted ensemble target model
  $\mathbf{x}$ Initial benign sample
  $\bar{o}$ Target output
  c Number of classes
Output
  $\bar{\mathbf{x}}$ Evading sample
$t \leftarrow \frac{1}{c}$
$\bar{\mathbf{x}} \leftarrow \mathbf{x}$
while True **do**
  $\bar{\mathbf{X}} \leftarrow [\,]$
  for $T \in E$ **do**
    $\bar{\mathbf{X}} \leftarrow \bar{\mathbf{X}} \cup$ evading candidates for T using $\bar{\mathbf{x}}$
  end for
  if $E(\bar{\mathbf{x}}) = \bar{\mathbf{o}}$ **for any** $\bar{\mathbf{x}} \in \bar{\mathbf{X}}$ **then**
    $\bar{\mathbf{X}} \leftarrow [\bar{\mathbf{x}} \in \bar{\mathbf{X}}$ **where** $E(\bar{\mathbf{x}}) = \bar{\mathbf{o}}]$
    return $\arg\min_{\bar{\mathbf{x}} \in \bar{\mathbf{X}}} \|\bar{\mathbf{x}} - \mathbf{x}\|$
  end if
  $\bar{\mathbf{x}} \leftarrow \arg\min_{\bar{\mathbf{x}} \in \bar{\mathbf{X}}} s(\mathbf{x}, \bar{\mathbf{x}}, E, t)$
end while

Evasion Attack Evaluation

- RT-IoT2022 (Binary Classification / Anomaly Detection)
  - 81 features
  - 9,266 training samples
  - Decision Tree, Random Forest, AdaBoost, Isolation Forest

Victim Model Performance

Model	F1-Score (%)
Decision Tree	98.88
Random Forest	99.34
Adaptive Boosting	99.35
Isolation Forest	85.71

Evaluation Results – RT-IoT2022

Anomaly to Normal Evasion Performance (1,000 samples) L_2 Normal

	Mean	Median	Min.	Max.	Std. Dev.
Decision Tree	0.0025751	0.0003695	0.0000039	0.1110265	0.0109706
Random Forest	0.0681997	0.0311600	0.0000034	0.9151371	0.1262921
Adaptive Boosting	0.0039809	0.0006478	0.0000013	0.1651765	0.0113164
Isolation Forest	5.6200075	1.9886899	0.0000079	438.0973418	16.5729001

Normal To Anomaly Evasion Performance (1,000 samples)

	Mean	Median	Min.	Max.	Std. Dev.
Decision Tree	0.0009272	0.0006455	0.0000182	0.0699293	0.0036668
Random Forest	0.0344304	0.0008061	0.0000853	0.01472549	0.0493456
Adaptive Boosting	0.0007171	0.0006693	0.0000099	0.0140307	0.0011364
Isolation Forest	6.5669925	2.6590401	0.0134395	63.3404823	10.0700819

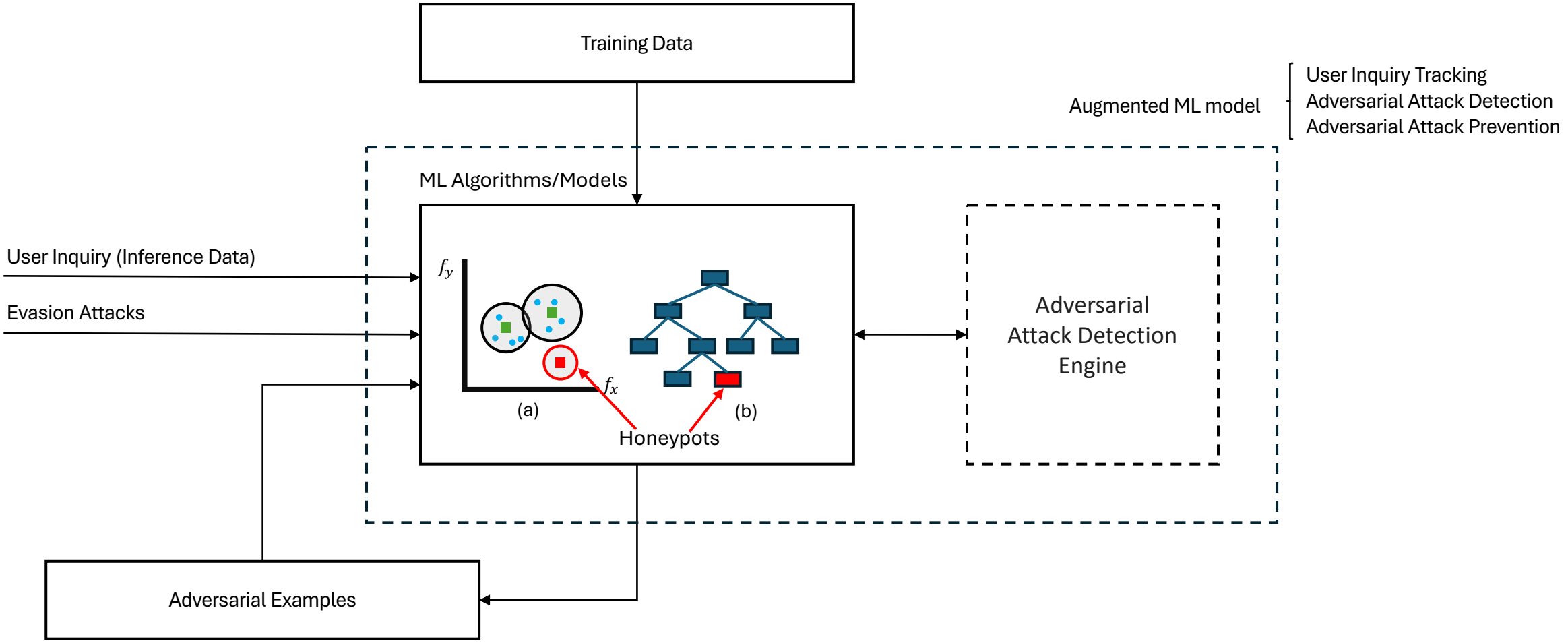
Defense Mechanisms

- Detection: identifying adversarial examples.
  - Adversarial training
  - Statistics test
- Prevention: using auxiliary components within the model to mitigate the impact of adversarial examples.
  - Ensemble learning
  - The GAN defense
- Risk assessment: defining trust boundaries for ML models and mitigate risks through cybersecurity strategies.

Defense Mechanisms Comparison

	Pros	Cons
Detection-based	Capable of identifying certain adversarial examples. Adversarial training enhances model robustness but is not a dedicated detection method.	Adversarial training requires the generation of adversarial examples for effective learning. Statistical tests require sufficiently large input sample sets for reliable detection. Effectiveness in real-time scenarios is questionable due to computational overhead. Detection methods may be specific to certain types of adversarial attacks and might not generalize well.
Prevention-based	Can potentially mitigate a broad range of adversarial attacks.	Often requires additional components or modifications to ML models, such as input preprocessing, feature squeezing, or gradient masking.
Risk assessment	May help identify and mitigate external attacks through monitoring and threat modeling.	Does not directly prevent specific adversarial attacks but aids in understanding vulnerabilities and potential risks.

A Honeypot-Based Approach for Detecting Adversarial Attacks in Machine Learning



ML Security Challenges

- Explainability challenge
- Undetectable and untraceable attacks
- Information exposure
  - black-box, white-box, and gray-box models
- Inadequate defense mechanisms
  - No reliable, robust defense to a wide range of adversarial attacks on ML

Summary

- Adversarial ML and attack taxonomy
- ML threat model
- Exploratory attack algorithm – hypersphere attack
  - Model inversion and parameter stealing
- Exploitative attack algorithm – data evasion attack
  - Decision Tree, Random Forest, Adaptive Boosting, Isolation Forest algorithms
  - Supervised and unsupervised
- Defense against adversarial attacks
  - Implement robust security measures when deploying ML models in adversarial environments.

References

1. Tesla, “AI & Robotics,” tesla.com. https://www.tesla.com/AI (accessed Nov. 20, 2024).
2. Sanford Health, “Sanford partners with Dandelion, Sharp on new data platform,” sanfordhealth.org. https://news.sanfordhealth.org/news-release/sanford-partners-with-dandelion-sharp-on-new-data-platform/ (accessed Nov. 20, 2024).
3. N. Carlini, M. Jagielski, C. A. Choquette-Choo, D. Paleka, W. Pearce, H. Anderson, A. Terzis, K. Thomas, and F. Tram`er, “Poisoning web-scale training datasets is practical,” in *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 176–176, IEEE Computer Society, 2024.
4. C. Koball, Y. Wang, B. P. Rimal and V. Vaidyan, "Machine Learning Security: Threat Model, Attacks, and Challenges," in *Computer*, vol. 57, no. 10, pp. 26-35, Oct. 2024, doi: 10.1109/MC.2024.3396357.
5. C. Koball, Y. Wang, Varghese Vaidyan, and John Hastings, “Assessing Evasion Attacks on Tree-Based Machine Learning Models: Supervised vs. Unsupervised Approaches”, in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 11-14, 2025.
6. C. Koball and Y. Wang, “Extracting Information from Hypersphere-Based Machine Learning Models”, in *IEEE Consumer Communications & Networking Conference (CCNC)*, 10–13 January 2025, Las Vegas, NV, USA.
7. A. Vassilev, A. Oprea, A. Fordyce and H. Anderson, Adversarial machine learning: A taxonomy and terminology of attacks and mitigations, 2024, doi: 10.6028/NIST.AI.100-2e2023.