QuantUniversity, LLC

www.quantuniversity.com

NIST's Adversarial Machine Learning

Module 2: Predictive AI Taxonomy

2. Predictive AI Taxonomy

2.1 Attack Classification

- Taxonomy of attacks in adversarial machine learning for PredAI systems provided. The objectives:
  - **Availability** breakdown
  - **Integrity** violations
  - **Privacy** compromise.
- Attackers employ certain capabilities to achieve these objectives.
- Individual attack classes and their required capabilities to achieve a specific objective are distinguished.

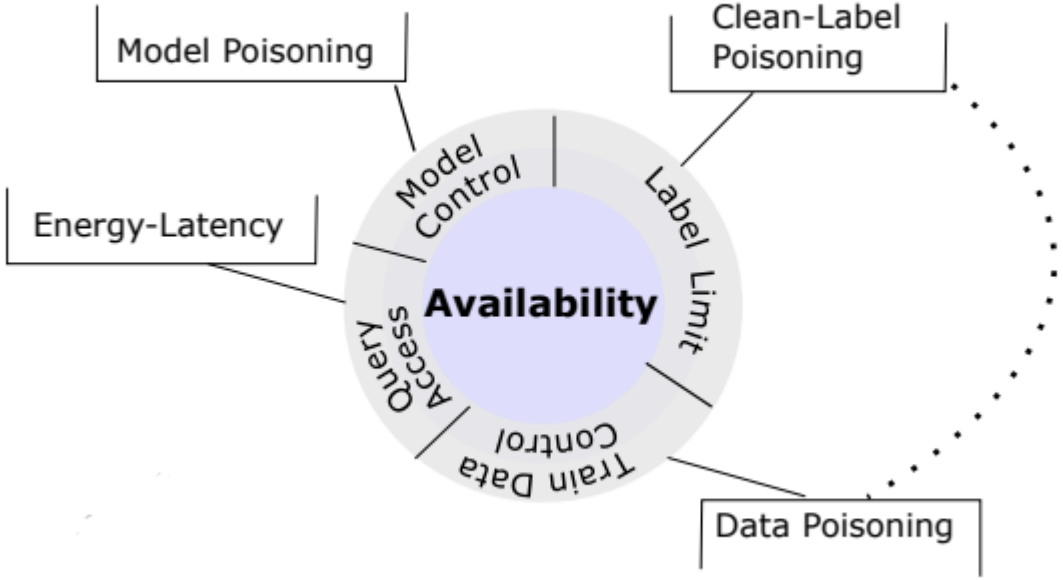
Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

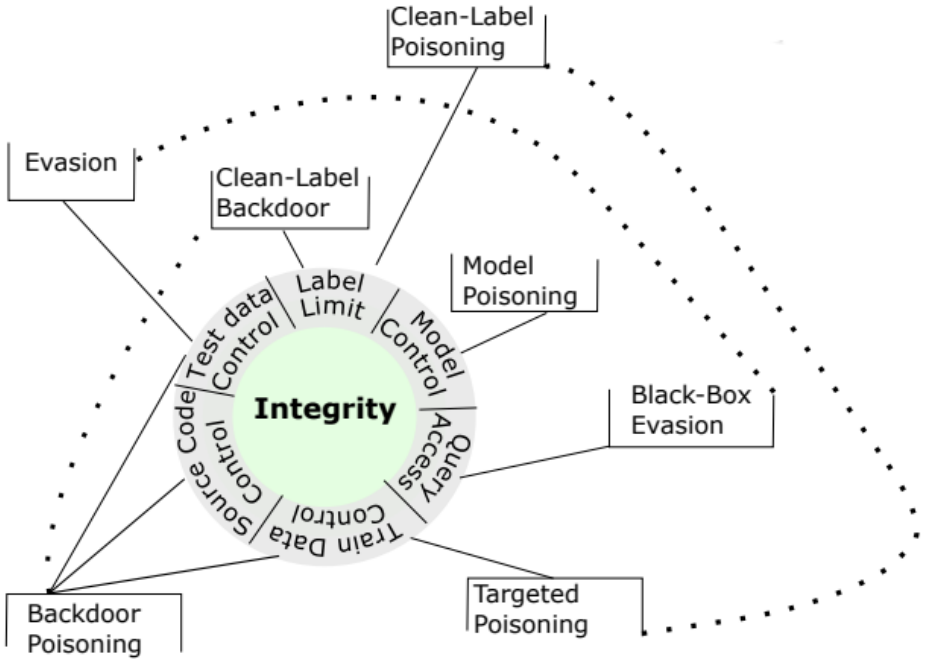
2. Predictive AI Taxonomy

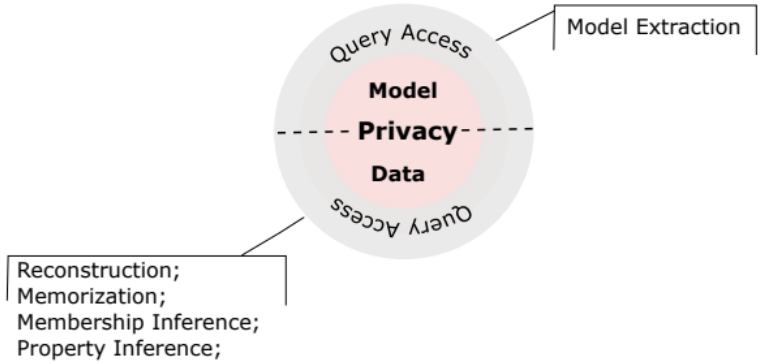
2.1 Attack Classification

- The **attack classifications** rely on various dimensions such as:
  - The learning method and stage when the attack is executed
  - Attacker's goals and objectives
  - Adversary's capabilities
  - Attacker's knowledge of the learning process
- The aim is to establish a standard **terminology** for adversarial attacks on machine learning, thus unifying all existing work.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf







Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.1 Stages of Learning

- Machine learning involves a **training stage**, in which a model is learned, and a **deployment stage**, in which the model is used to generate predictions.
- ML models may be **generative** or **discriminative**. PredAI models are usually discriminative.
- **Adversarial machine learning** considers potential attacks against AI systems at the training or deployment stage.
- The attacker might control parts of the training data, their labels, the model parameters, or the ML algorithms code.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.1 Stages of Learning

- **Training-time attacks** are called **poisoning** attacks.
  - In a **data poisoning** attack, an adversary controls a subset of the training data.
  - In a **model poisoning** attack, the adversary controls the model and its parameters.
- **Deployment-time attacks** include
  - **Evasion attacks** which create adversarial examples
  - **Privacy attacks** which infer sensitive information about the training data or the ML model.
- These could be further divided into **data privacy attacks** and **model privacy attacks**.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.2 Attacker Goals and Objectives

- Classifying attacker's objectives into three dimensions:
  - **Availability**
  - **Integrity**
  - **Confidentiality**
- Adversarial success indicates achieving one or more of these objectives.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.2 Attacker Goals and Objectives

- **Availability Breakdown**
  - It is an attack against ML attempting to break down the performance of the model at deployment time
  - Availability attack can be mounted via data poisoning, model poisoning, or as energy-latency attacks.
- **Integrity Violations**
  - It targets the integrity of an ML model's output, resulting in incorrect predictions
  - Integrity violation can be performed by mounting an evasion attack at deployment time or a poisoning attack at training time.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.2 Attacker Goals and Objectives

- **Privacy Compromise**
  - Attackers might be interested in learning information about the training data or about the ML model
  - Compromising the **privacy** of training data could include data reconstruction, membership-inference attacks, data extraction, property inference
  - Model extraction is a model privacy attack aiming to extract information about the model.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.3 Attacker Capabilities

- An adversary might use six types of **capabilities** to achieve objectives:
  - **Training data control**: Attackers can control a subset of the training data by inserting or modifying training samples for poisoning attacks.
  - **Model control**: Attackers may control model parameters, either by generating a Trojan trigger and inserting it in the model or by sending malicious local model updates in federated learning.
  - **Testing data control**: The attacker can add perturbations to testing samples at model deployment time for evasion attacks to generate adversarial examples or in backdoor poisoning attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.3 Attacker Capabilities

- **Source code control**: Attackers can modify the source code of the ML algorithm, such as the random number generator or any third-party libraries, often open source.
- **Label Limit**: Restricting adversarial control of labels of training samples in supervised learning.
- **Query Access:** Submitting queries directly to the model hosted on cloud.

- Even without the ability to modify training/testing data, **source code**, or model parameters, access is crucial for **white-box attacks**.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.3 Attacker Capabilities

- Each attack class connects with the required capabilities to mount the attack. For example:
  - **Backdoor attacks** causing integrity violations require control of training data and testing data to insert the backdoor pattern.
  - Backdoor attacks can also be activated via source code control, particularly when training is outsourced.
  - **Clean-label backdoor attacks** do not allow label control on poisoned samples, but needs the capabilities necessary for backdoor attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.4 Attacker Knowledge

- **White-box attacks**: The attacker has **full** knowledge about the ML system, including the data, architecture, and hyper-parameters.
- **Black-box attacks**: Assumes **minimal** knowledge about the ML system. The attacker may get query access to the model, but no further information about the model's training.
- **Gray-box attacks**: This type of attack captures adversarial knowledge between black-box and white-box attacks. The attacker may know the **model architecture** but not its parameters, or vice versa.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.5 Data Modality

- Adversarial attacks on ML can occur in a range of **data modalities** used in different application domains
  - **Image** data typically involves attacks with continuous domain advantage to apply direct optimization.
  - In the case of **text**, such as NLP, attacks range from evasion and poisoning to privacy breaches.
  - **Audio** systems and text generated from audio signals have experienced attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.1 Attack Classification

2.1.5 Data Modality

- **Video** comprehension models have shown increased capabilities on vision-and-language tasks but are also vulnerable to attacks.
- **Cybersecurity** modality experienced poisoning attacks, malware classification, PDF malware classification, and Android malicious app detection.
- **Tabular data** had a variety of attacks against ML models in finance, business, and healthcare applications.

One open challenge is testing and measuring the resilience of several **multimodal ML** against evasion, poisoning, and privacy attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2. Predictive AI Taxonomy:

2.2 Evasion Attacks and Mitigations

- Evasion attacks involve creating **adversarial examples.**
  - Adversarial examples were originally demonstrated in linear classifiers for spam filters and became more intriguing in the context of **image classification**.
- Effective methods for generating adversarial examples against linear models and neural networks used **gradient optimization** on an adversarial objective function.
- Adversarial examples are also applicable in **black-box settings** where attackers only have query access to the model. These include zeroth-order optimization, discrete optimization, Bayesian optimization and transferability.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2. Predictive AI Taxonomy:

2.2 Evasion Attacks and Mitigations

- **Mitigating** adversarial examples is a well-known challenge. In the past, defenses have been published but broken by more powerful attacks.
- Most promising methods of mitigation include **adversarial training** (using adversarial examples in training), certified techniques like **randomized smoothing** and **formal verification techniques**.
- But these methods have their own limitations.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.1 White-Box Evasion Attacks

- **Adversarial examples** generation is often pursued using optimization methods, where the target class can be either decided by the attacker (targeted attacks) or it can be any incorrect class (untargeted attacks).
- **L-BFGS,** binary classifiers with malicious and benign classes and differentiable discriminant functions, **Fast Gradient Sign Method (FGSM)** are some of the methods that have been employed, particularly for deep learning models.
- Notable attacks include the **DeepFool**, **Carlini-Wagner attack**, and **Projected Gradient Descent (PGD) attack**.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.1 White-Box Evasion Attacks

- **Universal evasion attacks** entail constructing small universal perturbations that can be added to most images, inducing misclassification.
- **Physically realizable attacks**, has been realized through eyeglass frame printing that can evade or impersonate facial recognition systems. Other examples include the **ShapeShifter** attack and an attack applying stickers to road signs.
- Apart from computer vision applications, audio, video, natural language processing (NLP), and cybersecurity have also seen development of adversarial examples to deceive machine learning classifiers.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.1 White-Box Evasion Attacks

- **Audio:** Targeted attacks on models that transcribe text from speech, produce an audio waveform similar to an existing one but transcribable to any text of the attacker's choosing.
- **Video:** Adversarial evasion attacks can perturb a small number of video frames (sparse attacks) or every frame in a video (dense attacks).
- **Natural Language Processing (NLP):** Adversarial examples must abide by text semantics.
- **Cybersecurity:** Constraints imposed by application semantics and feature representations in cybersecurity force adversarial examples to respect these boundaries. This has led to methods such as FENCE and others that utilize formal logic to learn feature space constraints.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.2 Black-Box Evasion Attacks

- Black-box evasion attacks occur when the adversary doesn't have prior knowledge of the model architecture or training data but can query the trained ML model on various data samples.
- These attacks fall into two categories:
    - **Score-based attacks**
    - **Decision-based attacks**
- The main challenge in creating adversarial examples in black-box settings is to reduce the **number of queries** to the ML models.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.3 Transferability of Attacks

- Analysis of evasion attacks and mitigations via **transferability of attacks** which generate adversarial attacks under restrictive threat models.
  - One technique involves training a **substitute ML model**, generating white-box adversarial attacks on this model, and then transferring the attacks to the target model.
  - Various methodologies involve different ways of training the substitute models such as score-based queries or training an ensemble of models without querying the target model.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.3 Transferability of Attacks

- The **attack transferability** phenomenon has been extensively studied, with research attempting to understand why adversarial examples transfer across models.
  - It has been observed that intersecting decision boundaries in both benign and adversarial dimensions in different models can contribute to better transferability.
  - Two main factors identified that contribute to attack transferability are the **intrinsic adversarial vulnerability** of the target model and the **complexity of the surrogate model** used to optimize the attack.
- The concept of **Expectation Over Transformation** aims to create adversarial examples that can withstand real-world image transformations such as angle and viewpoint changes.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.2 Evasion Attacks and Mitigations

2.2.4 Mitigations

- Evasion Attacks are difficult to counter due to the widespread occurrence of adversarial examples across different ML models and domains.
- Defenses against adversarial evasion attacks have been distinguished into three robust categories:
    - **Adversarial Training**
    - **Randomized Smoothing**
    - **Formal Verification**
- Despite the robustness, these defenses have their inherent trade-offs, such as higher computational costs and adversely affecting model accuracy.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2. Predictive AI Taxonomy

2.3 Poisoning Attacks and Mitigations

- Poisoning attacks pose a significant threat during the training stage of machine learning systems; they have been extensively studied across various application domains.
    - With a notable history in **cybersecurity**, the first known poisoning attack was used for worm signature generation in 2006.
    - Recent attention has been drawn to poisoning attacks in industrial applications. A Microsoft report highlighted these as the most critical vulnerability for deployed machine learning systems.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2. Predictive AI Taxonomy

2.3 Poisoning Attacks and Mitigations

- Various kinds of **poisoning attacks** could result in availability or integrity violations and involve wide-ranging adversarial capabilities.
    - Availability poisoning attacks result in indiscriminate degradation of the machine learning model on all samples.
    - Stealthier targeted and backdoor poisoning attacks cause integrity violations on specific target samples.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2. Predictive AI Taxonomy

2.3 Poisoning Attacks and Mitigations

- Poisoning attacks leverage several **adversarial capabilities** like data poisoning, model poisoning, label control, source code control, and test data control, leading to different subcategories.
- They are found in multi-faceted adversarial scenarios, such as white-box, gray-box, and black-box models.
- Discussion involves the threat of availability poisoning, targeted poisoning, backdoor poisoning, and model poisoning attacks, classified according to their adversarial objective.
- Techniques for mounting such attacks and existing mitigations along with their limitations are also discussed in depth.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.1 Availability Poisoning

- **Availability Poisoning** involves attacks against AI systems causing denial-of-service attacks; they were first seen in cybersecurity applications targeting worm signature generation and spam classifiers.
  - These attacks **mislead** the worm signature generation algorithm causing the **misclassification of spam emails**.
  - ML-based methods are used for detecting such attacks, but training data could be mimicked, and the learning process could be **poisoned**.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.1 Availability Poisoning

- **Label flipping**, a simple black-box poisoning attack, requires many poisoning samples for mounting an attack, improved by optimization-based poisonings first introduced against **Support Vector Machines**.
    - These require white-box access to the model and training data for generating poisoning samples.
- Clean-label poisoning attacks mode scenarios where variant files can be submitted to threat intelligence platforms and labeling is performed using anti-virus signatures; detected by **monitoring the standard performance metrics** of ML models such as precision, recall, accuracy, F1 scores, and area under the curve.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.1 Availability Poisoning

- **Training data sanitization:** They are designed to clean the training set, remove poisoned samples before ML training, and protect datasets with cybersecurity mechanisms.
    - Including the **Region of Non-Interest** method, which excludes samples that decrease the accuracy of the model when added, and label cleaning for label flipping attacks.
    - Others use **outlier detection methods and clustering methods** for identifying poisoned samples.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.1 Availability Poisoning

- **Robust training:** Modifies the ML training algorithm and performs robust training instead of regular training, obtaining certification against label flipping attacks.
    - Techniques from robust optimization are used, such as a trimmed loss function and the use of randomized smoothing for adding noise during training.
    - Predictions are generated via model voting, error calculation, and the use of lift measurement.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.2 Targeted Poisoning

- Targeted **poisoning attacks** induce a change in an ML model's prediction on a specific set of samples usually by manipulating the labeling function of training data.
    - Techniques such as influence functions, optimization based on feature collision, and ConvexPolytope and BullseyePolytope leverage clean-label setting for the attack.
    - Subpopulation poisoning attacks target samples from an entire subpopulation.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.2 Targeted Poisoning

- **Mitigations** for targeted poisoning attacks are challenging.
  - Use of cybersecurity mechanisms for dataset provenance and integrity attestation is advised.
  - Differential privacy (DP) is proposed as a defense but there's a trade-off between robustness and accuracy.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.3 Backdoor Poisoning

- **Backdoor Poisoning:** The first backdoor poisoning attack was conceived with BadNets.
- New techniques make the location of the trigger dynamic or introduce functional triggers that alter according to the input.
- **Mitigations:** The defense strategies for such attacks involve training data sanitization, trigger reconstruction, and model inspection and sanitization. However, recent semantic and functional backdoor triggers pose challenges to these approaches.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.3 Backdoor Poisoning

- **Advanced Mitigations:** Techniques like outlier detection in the latent feature space and activation clustering work by isolating backdoored samples in a separate cluster.
- Additional methods involve reconstructing the **backdoor trigger** or analysing the trained model to determine if it is poisoned. System sanitization can then be performed via pruning, retraining or fine-tuning.
- **Limitations and Future Research:** Current defense strategies often fail against clean-label backdoor poisoning instances on malware classifiers. The training stage of meta classifiers is also computationally intense.
- **Poison Forensics:** An added layer of defense in an ML system could be poison forensics, which helps in identifying the malicious training examples and tracing back the source of attack in the training set.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.3 Backdoor Poisoning

- **Early Attempts:** Early poisoning attacks were designed against worm signature generation and spam detectors.
- **Cybersecurity Mitigations:** In cybersecurity, an autoencoder-based intrusion detection system, assuming malicious poisoning attack instances were under 2%, was developed.
- **A Different Perspective:** A recent paper provides a unique viewpoint on backdoor mitigation; showing that backdoors could be concurrent with naturally occurring features in the data. Further work is needed to identify and **remove** the backdoor-triggering samples.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.4 Model Poisoning

- **Model Poisoning**: Involves altering ML models to insert harmful functionality, either by modifying the trained model directly or by manipulating local model updates sent to the server in a federated learning context.
    - Ex. TrojNN: Reverse engineers a trained neural network's trigger then re-trains the model and poisons it.
    - Attackers poison components/models provided by suppliers by embedding malicious code.
- **Availability Attacks**
- **Targeted Poisoning & Backdoor Attacks**

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.3 Poisoning Attacks and Mitigations

2.3.4 Model Poisoning

- **Byzantine-resilient aggregation rules**: Designed to identify and exclude harmful updates during server-side aggregation.
- **Gradient Clipping & Differential Privacy**: These techniques may somewhat mitigate model poisoning attacks but at a sacrifice of accuracy.
- **Model Inspection & Sanitization**: Useful strategies for specific model poisoning vulnerabilities like backdoor attacks.
- **Program Verification Techniques**: These approaches, used in fields such as cryptographic protocol verification, may be applicable but pose challenges due to randomness and non-deterministic behavior in ML algorithms.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2. Predictive AI Taxonomy

2.4 Privacy Attacks

- **Reconstruction attacks** aim to reverse engineer **private information** from aggregate data.
- **Membership-inference attacks** are a type of privacy violation where an adversary determines if a record was included in the dataset.
- Other notable privacy attacks include **model extraction attacks** (designed to extract information about an ML model such as its architecture or model parameters) and **property inference attacks** (aiming to extract global information about a training dataset).

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.1 Data Reconstruction

- **Data reconstruction attacks** are a major privacy concern as they recover individual data from aggregate information.
- **Model inversion attacks** reconstruct class representatives from an ML model's training data. This type of attack cannot directly reconstruct the model's training data.
- The ability to reconstruct training samples is partly due to neural networks' tendency to memorize training data. This **memorization** aspect is necessary to achieve almost optimal generalization error in ML.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.1 Data Reconstruction

- **Reconstructor networks** can recover a sample from a neural network model, assuming an adversary with information about all other training samples.
- A binary neural network classifier's training data can be reconstructed from model parameters.
- In **attribute inference** attacks, attackers extract a sensitive attribute of the training set, assuming partial knowledge about other features in the training data.
- Neural networks' ability to memorize datasets shows the necessity of memorization for high-accuracy learning.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.1 Data Reconstruction

- Neural networks can memorize randomly selected datasets
- Training labels' memorization is required to achieve almost optimal generalization error in ML.
- **Next-symbol prediction** and **cluster labelling** uses memorization for high-accuracy learning.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.2 Membership Inference

- **Membership inference attacks** relate to privacy concerns when releasing aggregate information or machine learning models trained on user data. These attacks can reveal private information about an individual and can be used for data extraction attacks.
- These attacks are predominantly executed against **deep neural networks** used for classification.
- The success of an attacker in membership inference is defined by a cryptographically inspired game, involving the interaction with a challenger, for determining if a target sample was used in training the queried ML model.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.2 Membership Inference

- **Loss-based attack** and the technique of shadow models are among the commonly used techniques for mounting membership inference attacks.
- Despite their complexity difference, both techniques offer similar precision at low false positive rates.
- Notable alternatives include **LiRA attack** which assumes Gaussian model logit distributions.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.2 Membership Inference

- Membership inference attacks are also designed under the **label-only threat model** wherein the adversary only has access to the predicted labels of the queried samples.
- Several public privacy libraries like the TensorFlow Privacy library and the ML Privacy Meter provide implementations of these attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.3 Model Extraction

- The **goal** of a model extraction attack is to extract information about the **model architecture and parameters** by submitting queries to the ML model trained by an MLaaS provider.
- The first model stealing attacks were shown on several different ML models, however, extracting exact models have shown to be **impossible**; functionally equivalent models can be obtained though.
- Exercise of **model extraction** usually acts as a step towards more potent attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.3 Model Extraction

- Techniques for executing model extraction attacks include:
  - Direct extraction based on the **mathematical formulation**.
  - Using **learning methods** for extraction like active learning for more efficient extraction of model weights, and reinforcement learning for a number of query reduction.
  - Use of **side channel information** for model extraction, like using electromagnetic side channels or rowhammer attacks for model extraction of complex neural network architectures.
- Prevention of model extraction can significantly mitigate downstream attacks that bank on the attacker possessing the knowledge of the model architecture and weights.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.4 Property Inference

- **Property inference attacks** involve an attacker learning global information about the training data distribution by interacting with an ML model.
- These attacks have been demonstrated on various ML models, including hidden Markov models, Support Vector Machines, neural networks, and more.
- Some studies have shown that poisoning the property of interest can lead to a more effective distinguishing test for property inference.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.4 Property Inference

- Property inference attacks have been carried out on a variety of ML models:
    - Hidden Markov models and Support Vector Machines
    - Feed-forward neural networks
    - Convolutional neural networks
    - Federated learning models
- Other susceptible models include **generative adversarial networks** and **graph neural networks**.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.5 Mitigations

- The discovery of **reconstruction attacks** against aggregate information led to the rigorous definition of **differential privacy** (DP).
- DP has been widely adopted due to several properties: **group privacy**, **post-processing**, and **composition**. Common DP mechanisms include the Gaussian mechanism, the Laplace mechanism, and the Exponential mechanism. DP-SGD is the most prominent DP algorithm for training ML models.
- DP offers mitigation against data reconstruction and **membership inference attacks**. However, it does not guarantee protection against model extraction attacks.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.5 Mitigations

- A key challenge of using DP is achieving a trade-off between privacy and utility, typically measured as accuracy for ML models.
- A recent promising approach is **privacy auditing**, which measures actual privacy guarantees and determines privacy lower bounds by mounting privacy attacks.
- Efficient methods for privacy auditing with training a single model include using multiple random data canaries and client canaries along with a cosine similarity test statistics to audit user-level private federated learning.

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

2.4 Privacy Attacks

2.4.5 Mitigations

- Other techniques to counter model extraction include limiting user queries to the model, detecting suspicious queries, or creating more robust architectures to prevent side channel attacks, although these can be circumvented by motivated attackers.
- A potential approach to mitigating privacy leakage is **machine unlearning**, a technique that allows users to request the removal of their data from a trained ML model.
- Machine unlearning methods can be exact (retraining the model) or approximate (updating the model parameters to remove unlearned records influence).

Source: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-2e2023.ipd.pdf

QuantUniversity, LLC

www.quantuniversity.com

Thank you!

Contact

Email: info@qusandbox.com

www.QuantUniversity.com